



Universidad
Carlos III de Madrid

PROYECTO FIN DE CARRERA

DESARROLLO DE UNA APLICACIÓN DE MLEARNING PARA APRENDIZAJE DE IDIOMAS

Autor: Quesada Varela, Borja

Tutor: Zarraonandia Ayo, Telmo

Leganés, septiembre de 2015

Título: DESARROLLO DE UNA APLICACIÓN DE MLEARNING PARA
APRENDIZAJE DE IDIOMAS

Autor:

Director:

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día __ de _____ de
20__ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de
Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Agradecimientos

Quiero dar mi agradecimiento a Telmo Zarraonandia que me ha animado y ayudado a terminar mis estudios dándome las pautas y recomendaciones necesarias para poder acabar el proyecto fin de carrera. Así como a todos los profesores que tuve durante la carrera que han sido claves en completar mi formación académica.

Durante la etapa universitaria no solo he conseguido algo tan importante como formarme académicamente, también algo igual o más importante conocer a dos verdaderos amigos que siempre están ahí para apoyarme en cualquier cosa que necesite, por eso quiero agradecerlos mucho haberos conocido y tenerlos como amigos: Adrián y Roberto.

También me gustaría agradecer este proyecto a una persona que siempre ha estado a mi lado, aconsejando positivamente y motivándome para que siguiera adelante y no me rindiera, muchas gracias Silvia.

Y por último y no menos importante agradecer a la persona que ha estado desde siempre conmigo cuando las cosas han ido bien y cuando han ido mal. Día tras día aconsejándome, cuidándome y educándome para poder ser quien soy y conseguir lo que he conseguido, este proyecto va dedicado a ti mamá.

Resumen

El propósito del proyecto final de carrera consiste en la elaboración de una aplicación web adaptada a dispositivos móviles en la que se permita al profesor gestionar el material didáctico a través de objetos flash y a los alumnos la posibilidad de acceder a este material desde cualquier lugar, permitirá consultar el progreso del curso del alumno y esto le motivará a para poder ir avanzando y llegar a completarlo al 100%.

La aplicación web se implementará a través del lenguaje de programación Java en su edición J2EE y usando la tecnología Java Server Pages (JSP) y Servlets que nos permitirán la creación de web dinámica, y a través de la librería jQuery Mobile ofreceremos una visibilidad al usuario de la aplicación adaptada al dispositivo móvil del que se acceda.

Palabras clave:

Aprendizaje interactivo, aplicación web, dispositivos móviles, reutilización, facilidad de uso.

Abstract

The aim of this Bachelor Thesis is the development of an e-learning web application with responsive design where the professor is able to manage all teaching material with flash objects. Students may access the application from anywhere, being able to check their progress through the course. This will motivate them to continue and move forward until they have achieved the 100% of the course.

The web application will be implemented in Java Enterprise Edition (J2EE), using dynamic web technologies such as Java Server Pages (JSP) and Servlets. JQuery Mobile library will add responsive design for a perfect visualization in any device.

Keywords:

Interactive learning, web application, mobile devices, reuse, usability.

Índice

| | |
|---|-----------|
| 1. CAPÍTULO 1: INTRODUCCIÓN Y OBJETIVOS | 2 |
| 1.1 Introducción | 3 |
| 1.2 Objetivos | 5 |
| 1.3 Fases del desarrollo | 6 |
| 1.4 Glosario de términos | 8 |
| 1.5 Estructura de la memoria | 10 |
| 2. CAPÍTULO 2: ESTADO DEL ARTE | 11 |
| 2.1 Introducción | 12 |
| 2.2 HTTP | 16 |
| 2.3 HTML | 18 |
| 2.4 Java (J2EE) | 20 |
| 2.4.1 Servlets | 22 |
| 2.4.2 Java Server Pages | 25 |
| 2.4.3 JDBC | 27 |
| 2.5 MySQL | 28 |
| 2.6 Apache Tomcat | 30 |
| 2.7 jQuery Mobile | 31 |
| 3. CAPÍTULO 3: ANÁLISIS DEL SISTEMA | 33 |
| 3.1 Descripción | 34 |
| 3.2 Definición de requisitos | 35 |
| 3.3 Requisitos funcionales | 36 |
| 3.4 Requisitos no funcionales | 43 |
| 3.5 Requisitos de usabilidad | 45 |
| 4. CAPÍTULO 4: DISEÑO E IMPLEMENTACIÓN | 47 |
| 4.1 Arquitectura del sistema | 48 |
| 4.2 Base de Datos | 49 |
| 4.3 Componentes de la aplicación | 52 |
| 5. CAPÍTULO 5: EVALUACIÓN Y PRUEBAS | 58 |
| 5.1 Entornos de pruebas | 59 |
| 5.2 Definición de casos de prueba | 61 |
| 5.3 Trazabilidad de casos de prueba y resultados | 70 |
| 6. CAPÍTULO 6: PLANIFICACIÓN Y PRESUPUESTO | 72 |
| 6.1 Planificación | 73 |
| 6.2 Costes del proyecto | 75 |
| 6.2.1 Costes de personal | 75 |
| 6.2.2 Costes de material | 76 |
| 6.2.3 Coste total del proyecto | 76 |

| | |
|---|------------|
| 7. CAPÍTULO 7: CONCLUSIONES Y LÍNEAS FUTURAS | 77 |
| 7.1 Conclusiones..... | 78 |
| 7.2 Líneas futuras | 79 |
| 8. CAPÍTULO 8: BIBLIOGRAFÍA | 80 |
| 8.1 Bibliografía..... | 81 |
| 9. CAPÍTULO 9: ANEXO I – MANUAL DE USUARIO..... | 82 |
| 9.1 Introducción | 83 |
| 9.2 Requisitos necesarios | 83 |
| 9.3 Acceso aplicación | 84 |
| 9.4 Funcionalidades: Alumno | 85 |
| 9.4.1 Datos de inicio de sesión | 86 |
| 9.4.2 Realizar curso | 87 |
| 9.4.3 Consultar progreso del cursoG | 93 |
| 9.4.4 Cambiar contraseña | 94 |
| 9.4.5 Desconectarse..... | 95 |
| 9.5 Funcionalidades: Profesor | 96 |
| 9.5.1 Añadir módulo..... | 101 |
| 9.5.2 Añadir sección | 102 |
| 9.5.3 Añadir curso | 103 |
| 9.5.4 Añadir objeto | 104 |
| 9.5.5 Activar/Desactivar componente | 107 |
| 9.5.6 Consultar progreso alumno..... | 108 |
| 9.5.7 Desconectar de la aplicación | 109 |

Índice de figuras

| | |
|---|-----|
| Ilustración 1: Fases del desarrollo | 6 |
| Ilustración 2: Blackboard | 13 |
| Ilustración 3: Moodle | 13 |
| Ilustración 4: HTML | 18 |
| Ilustración 5: J2EE | 20 |
| Ilustración 6: MySQL | 28 |
| Ilustración 7: Tomcat | 30 |
| Ilustración 8: jQuery Mobile | 31 |
| Ilustración 9: MVC | 48 |
| Ilustración 10: Esquema entidad relación | 49 |
| Ilustración 11: Modelo relacional | 50 |
| Ilustración 12: Aplicación en IOS - Safari | 60 |
| Ilustración 13: Diagrama Gantt - Fase 1 | 74 |
| Ilustración 14: Diagrama Gantt - Fase 2 | 74 |
| Ilustración 15: Acceso aplicación | 84 |
| Ilustración 16: Datos inicio sesión | 86 |
| Ilustración 17: Módulos | 87 |
| Ilustración 18: Secciones..... | 88 |
| Ilustración 19: Cursos | 89 |
| Ilustración 20: Objetos | 90 |
| Ilustración 21: Objeto Flash | 91 |
| Ilustración 22: Opciones | 92 |
| Ilustración 23: Progreso del curso | 93 |
| Ilustración 24: Cambio de contraseña | 94 |
| Ilustración 25: Cambio de contraseña completado | 95 |
| Ilustración 26: Gestión de la aplicación | 96 |
| Ilustración 27: Gestión módulo | 97 |
| Ilustración 28: Gestión sección | 98 |
| Ilustración 29: Gestión curso | 99 |
| Ilustración 30: Gestión objeto | 100 |
| Ilustración 31: Añadir módulo | 101 |
| Ilustración 32: Añadir sección | 102 |
| Ilustración 33: Añadir curso | 103 |

| | |
|---|-----|
| Ilustración 34: Añadir objeto | 104 |
| Ilustración 35: Añadir objeto - Subir fichero | 105 |
| Ilustración 36: Objeto añadido | 106 |
| Ilustración 37: Activar/Desactivar componente | 107 |
| Ilustración 38: Seleccionar alumno | 108 |
| Ilustración 39: Consulta progreso alumno | 109 |

Índice de tablas

| | |
|------------------------------------|----|
| Tabla 1: Glosario de Términos..... | 9 |
| Tabla 2: Servlet | 24 |
| Tabla 3: JSP - Scripting..... | 25 |
| Tabla 4: JSP - Directivas..... | 26 |
| Tabla 5: JSP - Acciones | 26 |
| Tabla 6: JDBC..... | 27 |
| Tabla 7: RF-001 | 36 |
| Tabla 8: RF-002 | 37 |
| Tabla 9: RF-003 | 37 |
| Tabla 10: RF-004 | 38 |
| Tabla 11: RF-005 | 38 |
| Tabla 12: RF-006 | 39 |
| Tabla 13: RF-007 | 39 |
| Tabla 14: RF-008 | 40 |
| Tabla 15: RF-009 | 40 |
| Tabla 16: RF-010 | 41 |
| Tabla 17: RF-011 | 41 |
| Tabla 18: RF-012 | 42 |
| Tabla 19: RF-013 | 42 |
| Tabla 20: RNF-001 | 43 |
| Tabla 21: RNF-002 | 43 |
| Tabla 22: RNF-003 | 44 |
| Tabla 23: RNF-004 | 44 |
| Tabla 24: RU-001..... | 45 |
| Tabla 25: RU-002..... | 45 |
| Tabla 26: RU-003..... | 46 |
| Tabla 27: CP-001 | 61 |
| Tabla 28: CP-002 | 62 |
| Tabla 29: CP-003 | 62 |

| | |
|---|----|
| Tabla 30: CP-004 | 63 |
| Tabla 31: CP-005 | 63 |
| Tabla 32: CP-006 | 64 |
| Tabla 33: CP-007 | 64 |
| Tabla 34: CP-008 | 65 |
| Tabla 35: CP-009 | 65 |
| Tabla 36: CP-010 | 66 |
| Tabla 37: CP-011 | 66 |
| Tabla 38: CP-012 | 67 |
| Tabla 39: CP-013 | 67 |
| Tabla 40: CP-014 | 68 |
| Tabla 41: CP-015 | 68 |
| Tabla 42: CP-016 | 69 |
| Tabla 43: Matriz de trazabilidad | 71 |
| Tabla 44: Planificación - actividades | 73 |
| Tabla 45: Coste de personal | 75 |
| Tabla 46: Coste de material..... | 76 |
| Tabla 47: Coste total | 76 |

Capítulo 1

Introducción y objetivos

1.1 Introducción

En este apartado se describirá brevemente las motivaciones, los objetivos planteados, fases de desarrollo y la estructura de la memoria que se ha llevado a cabo para describir el proyecto fin de carrera.

La principal motivación para la realización del proyecto fin de carrera es que alumnos con necesidad de estudiar un idioma, en este caso inglés y que no disponen de suficiente tiempo para poder hacerlo, puedan desde su móvil, *tablet* u otro dispositivo móvil que dispongan, acceder al contenido de un curso tanto teoría, como ejercicios.

Otro punto a destacar para la realización de este proyecto es que ya la mayoría de alumnos y profesores, y en general la sociedad hace uso de internet a través del móvil. Según el estudio “Uso de Internet en Movilidad - 2015” realizado por Asociación para la Investigación de Medios de Comunicación (AIMC): el 85% de los encuestados declara haber utilizado un *smartphone* para conectarse a internet.

Así cuando el alumno disponga de algo de tiempo ya sea en transporte público, trabajo, universidad, casa o cualquier lugar en general y quiera acceder al curso pueda hacerlo sin necesidad de tener un ordenador delante, simplemente con su móvil y conexión a internet.

También esta visión la hemos trasladado a la parte del educador para que de igual manera pueda publicar el contenido de los cursos a través de su dispositivo móvil, desde cualquier lugar de forma rápida.

Es decir, el proyecto se ha basado en la metodología de enseñanza *M-learning* (aprendizaje electrónico móvil) que es una variante del *E-learning* (aprendizaje electrónico) pero orientado a dispositivos de conectividad inalámbrica y que para nuestro proyecto tiene las siguientes ventajas e inconvenientes:

Ventajas:

- Adaptación a las necesidades en tiempo y lugar del alumno/profesor.
- El alumno dispone de material para el aprendizaje: teoría, audios, ejercicios...
- No existe limitación de acceso como pudiera ser un curso presencial.
- Permite al profesor actualización de contenidos de forma sencilla.

CAPÍTULO 1: INTRODUCCIÓN Y OBJETIVOS

- Permite al profesor el acceso a estadísticas del progreso del curso de sus alumnos.

Inconvenientes:

- El usuario (alumno o profesor) de la aplicación web debe disponer de algún dispositivo móvil y conexión a internet.
- Requiere por parte del alumno más autonomía a la hora de realizar el curso.
- Requiere de un trabajo inicial mayor por parte del profesor en preparar todo el material del curso en formato electrónico para su posterior publicación.

Por tanto, el proyecto se ha basado en el análisis, diseño y desarrollo de una aplicación Web que permita el acceso a cursos de idiomas u otros materiales educativos desde dispositivos móviles.

La aplicación está dividida en dos principales vistas dependiendo del rol del usuario (alumnos y profesores). Los alumnos principalmente tendrán acceso al material educativo, estadísticas y otras opciones de gestión de su propio usuario. El profesor podrá gestionar los contenidos didácticos y consultar la información relativa al progreso de los cursos por parte de los alumnos.

La aplicación se encuentra implementada a través del lenguaje de programación Java en su edición J2EE y usando la tecnología Java Server Pages (JSP) y Servlets que nos permiten la creación de web dinámica.

Para el almacenamiento de datos se ha usado el sistema de gestión de bases de datos relacional MySQL y el servidor Web utilizado es Apache Tomcat.

Se han usado librerías específicas como jQuery Mobile, con el fin de que la compatibilidad en navegadores de dispositivos móviles sea óptima y su visualización correcta y agradable al usuario.

1.2 Objetivos

El objetivo principal que persigue el proyecto es el desarrollo del de una aplicación web que permita el acceso a material educativo desde dispositivos móvil, en particular a cursos de idiomas para la preparación de la certificación del nivel intermedio B1 de inglés.

Como objetivos más particulares que debe cumplir la aplicación:

- Acceso a material educativo por parte de los alumnos, con un diseño de un entorno amigable y de fácil navegación.
- Gestión a través de la aplicación de cursos por parte de profesores, que sea administrable, facilitando así el trabajo de los profesores.
- Desarrollo de una aplicación robusta que garantice la disponibilidad de los cursos e información relacionada para los alumnos y profesores
- Compatibilidad con la mayoría de los navegadores de dispositivos móviles, y permitiendo tanto a alumnos como a profesores trabajar desde cualquier lugar.
- Permitir la consulta por parte de los alumnos del progreso del curso y por parte del profesor de los cursos completados por cada uno de los alumnos.

1.3 Fases del desarrollo

En este apartado se definirá el proceso de ciclo de vida de la aplicación, estará compuesto por las siguientes fases:

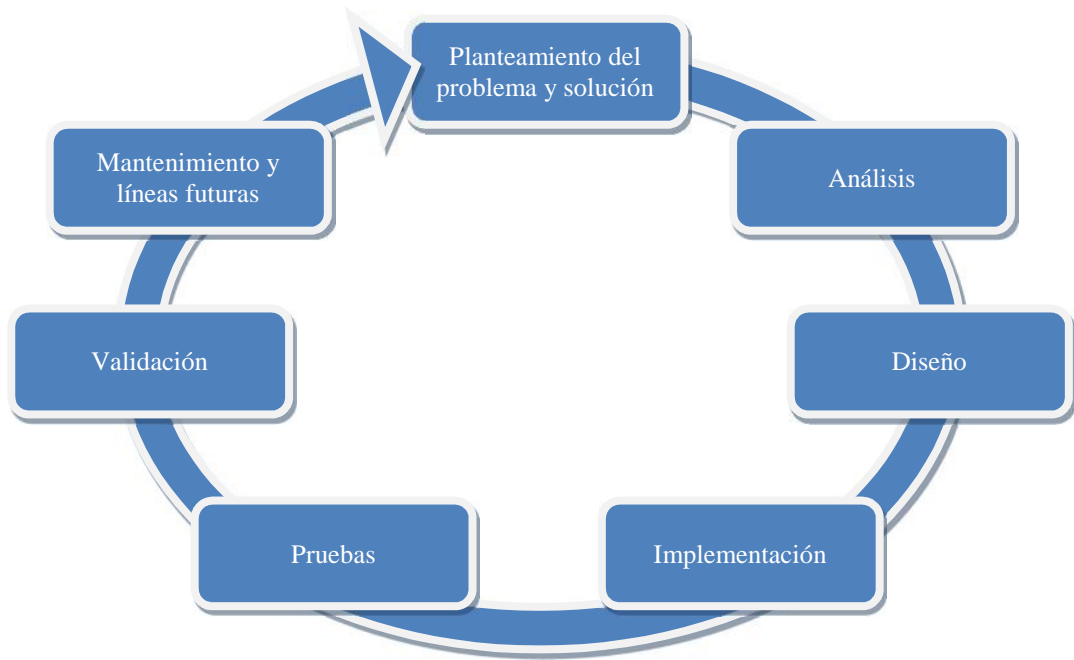


Ilustración 1: Fases del desarrollo

- Planteamiento del problema y solución. Se establece el punto de partida, se especifica qué buscamos y las soluciones que se dispondrán para llevarlo a efecto.
- Análisis. Se fijan los objetivos del sistema: estructura y funcionamiento, marcar las directrices que permitan alcanzar los objetivos propuestos. Se realiza una identificación de necesidades y requisitos del cliente y otros puntos que pueden ayudar a la identificación y desarrollo del proyecto.
- Diseño. Es la fase de diseño interno. Consiste en definir cómo organizar las necesidades de forma adecuada para la ejecución. Posteriormente se lleva a cabo un diseño detallado donde se describen las especificaciones de la aplicación.

CAPÍTULO 1: INTRODUCCIÓN Y OBJETIVOS

- Implementación. Es la fase de programación. Traslado del diseño a código. La fase con mayor duración, no se trata de una etapa a realizar de manera única, sino que está abierta al cambio.
- Pruebas. En esta etapa se realiza el diseño de casos de prueba con el fin de asegurar el correcto funcionamiento de la aplicación.
- Validación. Proceso de mejora y optimización del software después de su entrega al usuario final (es decir; revisión del programa), así como también corrección de los defectos.
- Mantenimiento y ampliaciones futuras. Involucra cambios en el software con el fin de corregir defectos y dependencias encontradas durante su uso. Posibles nuevas funcionalidades y hacer reusable la aplicación para otros materiales didácticos que se deseen publicar para la utilización de los alumnos de la universidad.

1.4 Glosario de términos

En este apartado se recopilan los términos más importantes que se hayan utilizado a lo largo del documento:

| Término | Definición |
|---------------------|--|
| AIMC | Asociación para la investigación de medios de comunicación |
| AIX | Advanced Interactive eXecutive, sistema operativo de IBM |
| API | Interfaz de Programación de Aplicaciones |
| BBDD | Base de datos |
| BSD | Berkeley Software Distribution, sistema operativo |
| CMS | Sistema de gestión de contenidos |
| CP | Caso de prueba |
| DAO | Objeto de Acceso a Datos |
| DHTMLX | Librería de JavaScript |
| EJB | Enterprise JavaBeans |
| EL | Expression Language |
| HTML | Lenguaje de marcas de hipertexto |
| HTTP | Protocolo de transferencia de hipertexto. |
| IOS | Sistema operativo móvil de Apple Inc. |
| J2EE | Java 2 Enterprise Edition |
| J2SE | Java Platform Standard Edition |
| JDBC | Java Database Connectivity |
| jQueryMobile | Librería de Javascript |
| JSF | JavaServer Faces |
| JSP | JavaServer Pages. |
| JSTL | Librería de Etiquetas Estándar de JavaServer Pages |
| LMS | Sistema de gestión de aprendizaje |
| MIME | Extensiones multipropósito de correo de internet |
| MVC | Modelo Vista Controlador |
| MyISAM | mecanismo de almacenamiento de datos de MySQL |
| MySQL | Es un sistema de gestión de bases de datos |
| ODBC | Open DataBase Connectivity |
| PHP | Lenguaje de programación |
| RF | Requisito funcional |
| RNF | Requisito no funcional |
| RU | Requisito de usabilidad |

CAPÍTULO 1: INTRODUCCIÓN Y OBJETIVOS

| Término | Definición |
|-------------|--------------------------------------|
| SGBD | Sistema de gestión de base de datos |
| SQL | Structured Query Language |
| TCL | Lenguaje de herramientas de comando |
| TCP | Protocolo de Control de Transmisión. |
| W3C | Consortio para World Wide Web |

Tabla 1: Glosario de Términos

1.5 Estructura de la memoria

El presente documento se ha estructurado en 9 capítulos que se detallan a continuación:

- Capítulo 1. Introducción: Describe los objetivos del proyecto, entorno, herramientas, etc. Introduciendo al lector en la cuestión que trata el proyecto. Se expone el ciclo de vida del producto software a desarrollar y los recursos necesarios. En resumen, se pretende dar una visión inicial y global del proyecto fin de carrera.
- Capítulo 2. Estado del arte: Describe en profundidad lo que existe en el momento de realizar el proyecto sobre el tema que aborda, tecnología y otras alternativas disponibles.
- Capítulo 3. Análisis del sistema: tiene el propósito de producir y analizar los requisitos del usuario y los requisitos técnicos del sistema.
- Capítulo 4. Diseño e implementación: En este capítulo se explica la solución desarrollada, el diseño detallado de la base de datos utilizada, algunos aspectos interesantes del proceso de implementación y consideraciones a tener en cuenta.
- Capítulo 5. Evaluación y pruebas: se realizará una batería de pruebas realizada al sistema en su totalidad y que permitirá validar los requisitos identificados.
- Capítulo 6. Planificación y presupuesto: se expondrá la planificación detallada que se seguirá en el desarrollo del trabajo, así como el presupuesto, que se explicará por partes, justificando los costes de personal, hardware y software, e informando del coste total del trabajo, incluyendo beneficios y margen de riesgo.
- Capítulo 7. Conclusiones y líneas futuras: Este apartado contiene las impresiones obtenidas de la realización del proyecto y así como una lista de posibles aspectos ampliables en versiones posteriores
- Capítulo 8. Bibliografía, se mostrarán todas las fuentes de información consultadas durante el proceso de desarrollo del trabajo.
- Capítulo 9. Anexo 1: Manual de usuario, se detallará técnicamente como usar la aplicación.

Capítulo 2

Estado del Arte

2.1 Introducción

El este capítulo se expondrá tanto las aplicaciones parecidas que actualmente existen, como las tecnologías y herramientas utilizadas para implementar la aplicación Web “B1 Certificate Trainer”.

Desde un punto de vista estructural la aplicación web desarrollada es un software tipo CMS (sistema gestión de contenidos) que genera las páginas de manera dinámica. Nuestro desarrollo al estar centrado en la gestión de contenidos didácticos podemos decir que está encasillado en el tipo de software llamado LMS (sistema de gestión de aprendizaje).

Los sistemas de administración del aprendizaje, o administración de cursos o contenido académico (LMS, Learning Management Systems) son herramientas computacionales, generalmente basadas en la web, que proveen interacción entre uno o más docentes con sus estudiantes. Estas herramientas llamadas también plataformas tecnológicas sirven como medio de apoyo en diversas modalidades educativas ya que la gestión de los contenidos que alberga facilita la utilización de éstos sin restricciones de horarios o lugares físicos en lo que se tenga que coincidir para poder trabajar.

Los LMS ofrecen las siguientes ventajas:

- Posibilidades de formación flexible y económica.
- Anulan las distancias geográficas y temporales.
- Combinan el potencial de Internet con el de las herramientas tecnológicas educativas.
- Libertad en tiempo y ritmo de aprendizaje.
- Aprendizaje constante y nutrido a través de la interacción entre docentes y alumnos.
- Uso de plataformas con mínimos conocimientos.

En la actualidad las aplicaciones del tipo LMS más importantes por su uso son:

- Blackboard: es un sistema de aprendizaje y administrador de contenidos y cursos en línea, propiedad de Blackboard Inc., empresa fundada en Washington. D.C. en 1997. A la fecha Blackboard ofrece una gama de productos de software para administrar aprendizaje en línea (*e-learning*), procesamiento de transacciones, comercio electrónico y manejo de comunidades en línea. Sus productos de software propietario son utilizados por miles de instituciones educativas y empresas de todo el mundo.



Ilustración 2: Blackboard

- Moodle: es sistema de gestión de contenidos y cursos de distribución libre creado por Martin Dougiamas, quien fuera administrador de WebCT en la Universidad Tecnológica de Curtin, un sistema comercial de aprendizaje virtual que se fusionó en 2005 con Blackboard, Inc. Dougiamas basó su diseño en las ideas del constructivismo en pedagogía.



Ilustración 3: Moodle

Ambos son dos sistemas de administración de aprendizaje muy completos, pero quizá su diferencia más notable es la licencia de uso. Blackboard es software propietario y sus licencias llegan a costar miles de dólares, mientras que Moodle es software libre, el cual es completamente gratuito que opera con una licencia GPL (*General Public Licence*).

Otras aplicaciones del tipo LMS que existen en la actualidad son:

- Claroline: La comunicación está dada por foros, correos electrónicos y chat. En cuestión de evaluaciones el sistema proporciona una herramienta de seguimiento del alumno, las calificaciones son oportunas y al instante.
- Dokeos: Se cuenta con chat y foros de manera muy efectiva, cada alumno puede subir su fotografía para que el trato sea menos frío en este ambiente virtual. El sistema de evaluación es eficiente ya que existe un seguimiento por alumno, el sistema califica en tiempo real y ofrece retroalimentación en test y cuestionarios.
- Ilias: La comunicación se establece por correos electrónicos, chats y foros. Las calificaciones son oportunamente publicadas en línea y existe también un seguimiento de alumnos, pero es una navegación un tanto confusa para el usuario y como consecuencia, no hay fluidez en los recorridos por el sitio.
- Eledge Open Learning Management System: Este software, desarrollado por la Universidad de Utah, es un entorno de creación de un sitio web para la educación en línea que incluye el registro de estudiantes, la autenticación, la creación de contenido, encuestas, exámenes, evaluación de trabajos, informes, libro de notas del instructor, calendario de clase y ayuda en línea.

Sus características principales son:

- Capacidad de subir archivos en varios formatos (HTML y formato binario).
- Posibilidad de gestionar los contenidos de los cursos.
- Los alumnos tienen la posibilidad de matricularse a un curso y hay una funcionalidad de seguimiento del alumno.

Como conclusión los LMS son otro medio de comunicación más para fomentar y reforzar el proceso enseñanza aprendizaje en los estudiantes y teniendo en cuenta que las tecnologías han sido desarrolladas para acercarnos cada vez más, para facilitarnos tareas, para apoyarnos en nuestras actividades y no para desplazarnos del mundo de la educación.

En estas aplicaciones la funcionalidad es muy similar a la que hemos implementado en nuestro proyecto. Así, todas ellas permiten al administrador la gestión de los cursos, obtener información sobre el avance por parte de los alumnos, incluyen herramientas de evaluación de los cursos, ejercicios, audios y otras prácticas.

Para la implementación del código de la aplicación “B1 Certificate Trainer” se ha usado como herramienta de programación Eclipse y la aplicación está desarrollada mediante código en Java (J2EE): Servlets y JSP conectando a una base de datos MySQL y desplegada bajo un servidor Apache Tomcat.

En el desarrollo se ha puesto especial hincapié en que la aplicación Web permita un perfecto funcionamiento en dispositivos móviles, para ello se ha usado la librería jQuery Mobile.

2.2 HTTP

HTTP (Protocolo de Transferencia de Hipertexto) es un protocolo estático, el servidor no guarda ninguna información sobre el cliente una vez enviada la respuesta. El intercambio de información entre el cliente y el servidor se realiza mediante una petición y una respuesta (request / response)

Para realizar una petición se especifica una URL (*Uniform Resource Locator*), un ejemplo sería:

<http://localhost:8080/B1training/Acceso.jsp>

- La primera parte (http://) especifica que la petición utiliza el protocolo HTTP.
- A continuación, se indica el nombre del servidor. Por defecto el puerto es el 80, si se necesita usar otro puerto se deberá especificar después de los dos puntos (localhost:8080)
- Y la última parte indica el recurso que se llama dentro del servidor, (B1training/Acceso.jsp)

En la petición HTTP Request, hay dos métodos principales para realizar las peticiones:

- GET: No necesita ningún tipo de proceso en el servidor. Generalmente utilizado para recuperar recursos estáticos y en forma de texto. El paso de parámetros es dentro de la URL, como texto, con el formato:
 - URL?nombreParámetro=valor&nombre=valor
- POST: Necesita algún tipo de proceso. Para recuperar recursos dinámicos o formatos no de texto. El paso de parámetros se envía como parte del *body*.

La respuesta, HTTP Response:

- Está compuesto de una cabecera (donde se informa el estado de la respuesta) y un cuerpo (la información solicitada)
- Utiliza MIME (*Multi-Purpose Internet Mail Extension*) para indicar el tipo de contenido en la petición y en la respuesta. Ejemplos MIME: text/html, image/gif, application/x-shockwave-flash, etc.

2.3 HTML

HTML (lenguaje de marcas de hipertexto), es la base del código usado en las JSP de la aplicación, es un lenguaje de marcado que se basa en dos principales elementos:

- Etiquetas: donde existe la etiqueta de apertura y de cierre y su función es marcar las diferentes partes de la página web: texto, imágenes, videos, objetos flash u otros.
- Atributos: estos se incluyen dentro de la etiqueta de apertura y su funcionalidad es muy variada, desde cambiar el estilo del contenido de la etiqueta, mostrar el contenido en varios idiomas o lanzar un evento para realizar acciones dinámicas (estos últimos atributos son los más usados las JSP implementadas).

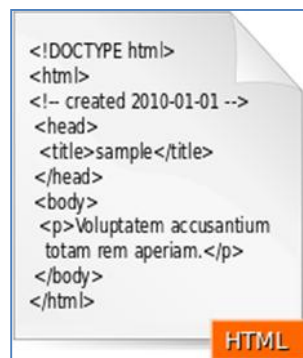


Ilustración 4: HTML

A nivel estructural el código html que está insertado en las JSP se compone como todo código HTML:

- Cabecera: se encuentra entre las etiquetas `<head>` y `</head>`, en esta parte del código la información no es visible al usuario excepto el título que está delimitado entre las etiquetas `<title>` y `</title>`.
- Cuerpo: en esta zona del código está la parte visible al usuario: texto, imágenes y en nuestra aplicación objetos flash embebidos (se describirá en el apartado de diseño e implementación). Este código estará situado entre las etiquetas de apertura `<body>` y cierre `</body>`.

Y a su vez tanto la cabecera como el cuerpo están comprendidas entre las etiquetas `<html>` y `</html>`

Este lenguaje sencillo de escribir y regulado por el Consorcio W3C, pero no nos proporciona un medio para hacer dinámica nuestra aplicación web.

Por ello utilizaremos Java Server Pages, haremos uso de código HTML dentro de las propias JSP que describiremos posteriormente.

2.4 Java (J2EE)

J2EE es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en lenguaje de programación Java con una arquitectura de programación distribuida de N niveles (Multicapa).

Las aplicaciones Java EE están creadas a partir de componentes. Un componente Java EE es un objeto de software con su propia funcionalidad que está incluido dentro de una aplicación Java EE con sus ficheros y clases y es capaz de comunicarse con otros componentes de la aplicación.

Los tipos de componentes Java EE son:

- Aplicaciones que se ejecutan en el cliente
- Servlets, JSF y JSP componentes web que se ejecutan en el servidor (.war)
- Componentes de negocio que se ejecutan en el servidor (.jar)



Ilustración 5: J2EE

Java EE incluye varias especificaciones de API y define cómo usarlas. A parte incluye configuraciones específicas para componentes únicas en Java EE: EJB, Servlets, Portlets, JSP, etc. En nuestra aplicación hemos hecho uso de Servlets y JSP.

Las características principales de la programación bajo la plataforma J2EE son las siguientes:

- Portable entre plataformas, menor dependencia del software entre distintas plataformas.
- Es escalable, es decir se adapta correctamente el software desarrollado a las distintas configuraciones.
- Existencia de herramientas de código abierto disponibles para hacer extensible la aplicación.
- Reducción del tiempo de desarrollo y costes.
- Reducción de la complejidad de las aplicaciones y por consiguiente mayor facilidad en su mantenimiento.
- Incremento del rendimiento de la aplicación.

Los componentes Java EE son desplegados en un servidor de aplicaciones, en el caso de nuestra aplicación sobre un servidor Apache Tomcat (véase apartado 2.6).

Este servidor será el encargado de ejecutar y manejar dichos componentes. Al desplegar una aplicación, el servidor hace todos los chequeos necesarios.

2.4.1 Servlets

Los *servlet* son clases java que responden dinámicamente a peticiones realizadas por el usuario y construyen una respuesta.

Los *servlets* son más apropiados para gestionar los datos no visibles al usuario como conexiones, actualización de datos y obtención de los mismos para posteriormente generar la vista que se le mostrará al usuario. Y las JSP son más apropiadas para generar las vistas dinámicas. Así aplicamos en nuestro proyecto el modelo Vista-Controlador.

La tecnología Java *Servlet* es el fundamento de toda la tecnología de aplicaciones web, añade un nivel de abstracción que hace que el desarrollo de aplicaciones web y de prototipos sea más rápido, y que la creación de aplicaciones web en sí, sea más fácil de mantener, escalable y robusta.

Los paquetes que forman parte son:

- `javax.servlet`
- `javax.servlet.http`

| Propósito | Clase/Interface | Descripción |
|-----------------------------------|--|---|
| Implementación de servlets | <code>javax.servlet.Servlet</code> <code>javax.servlet.SingleThreadModel</code> <code>javax.servlet.GenericServlet</code> <code>javax.servlet.http.HttpServlet</code> | Las clases/interfaces bajo esta categoría tienen la finalidad de permitir implementar <i>servlets</i> . |

| Propósito | Clase/Interface | Descripción |
|----------------------------------|---|---|
| Configuración del servlet | javax.servlet.ServletConfig | La API proporciona varios medios para acceder al objeto <i>ServletConfig</i> que nos da acceso a ciertos parámetros de inicialización que pueden ser configurados en fase de despliegue. |
| Servlet Exceptions | javax.servlet.ServletException javax.servlet.UnavailableException | Estas excepciones son lanzadas por el <i>servlet</i> para que se realice el correspondiente tratamiento de errores. |
| Peticiones y Respuestas | javax.servlet.ServletRequest javax.servlet.ServletResponse javax.servlet.ServletInputStream javax.servlet.ServletOutputStream javax.servlet.http.HttpServletRequest javax.servlet.http.HttpServletResponse | Estos objetos nos hacen posible poder leer datos de la entrada, y escribir la respuesta para el usuario. |
| Gestión de la sesión | javax.servlet.http.HttpSession javax.servlet.http.HttpSessionBindingListener javax.servlet.http.HttpSessionBindingEvent | En esta clase se guarda la información que se compartirá entre distintos <i>servlets</i> , enviando un objeto de este tipo en cada petición y respuesta: los datos del usuario conectado a la aplicación. |

| Propósito | Clase/Interface | Descripción |
|------------------------------------|--|---|
| Contexto del Servlet | javax.servlet.ServletContext | Permite compartir datos entre <i>servlets</i> de la misma aplicación web, se almacenan datos estáticos a diferencia que en la sesión. |
| Colaboración entre Servlets | javax.servlet.RequestDispatcher | Permite al <i>servlet</i> invocar a otro <i>servlet</i> , una JSP o incluso una página HTML, de esta forma mostramos las vistas al usuario. |
| Otros | javax.servlet.http.Cookie | Almacenan información de texto en el navegador del cliente. |

Tabla 2: Servlet

El *servlet* se ejecuta en un contenedor web, el cual es responsable del ciclo de vida del *servlet*, el ciclo de vida de un *servlet* se compone de los siguientes estados:

- **Instanciación:** El contenedor web crea una instancia del *servlet*.
- **Inicialización:** El contenedor invoca el método `init()`.
- **Servicio:** Si el contenedor recibe una petición para el *servlet*, invoca una instancia del método `service()` implementado en el *servlet*.
- **Destrucción:** Antes de eliminar la instancia del *servlet*, el contenedor invoca el método `destroy()`.
- **Inexistente:** La instancia es destruida y permanece a la espera de ser eliminada por el recolector de basura.

2.4.2 Java Server Pages

Java Server Pages es una tecnología para desarrollar páginas web que incluyen contenido dinámico. Se definen los archivos con la extensión *.jsp*.

En la aplicación web desarrolladas las JSP son las propias vistas de nuestro modelo de ingeniería del software MVC (Modelo Vista-Controlador)

Se componen de varios elementos, a destacar:

- Scripting: permiten incrustar código java dentro de código HTML. El código introducido se coloca en la misma posición en el código del *servlet* generado.

| Scripting | Código | Descripción |
|--------------------|-------------------------------------|---|
| Expresiones | <code><%= expresión %></code> | Código java de los que nos interesa mostrar su resultado en la página creada. En nuestra aplicación nos será útil para mostrar el porcentaje de curso completado por el alumno. |
| Scriptlet | <code><% código %></code> | Código java incrustado en un determinado punto del HTML. Nos permitirá obtener información en la propia <i>.jsp</i> . |
| Declaración | <code><%! código %></code> | Permite declarar métodos o variables de instancia. |

Tabla 3: JSP - Scripting

- Directivas: permiten realizar modificaciones al código del *servlet* de forma global, por ejemplo, importar librerías java externas o incluir el código fuente de otro JSP en el actual.

| Código | Descripción |
|---------------------------------------|---|
| <code><%@ page ... %></code> | Define atributos de la página. |
| <code><%@ include ... %></code> | Incluye un fichero durante la fase de traducción del JSP. |

Tabla 4: JSP - Directivas

- Acciones: Permiten controlar el flujo de la aplicación llamando a otros *servlets* o páginas JSP.

| Código | Descripción |
|----------------------------------|---|
| <code><jsp:include></code> | Incluye la respuesta de otro <i>servlet</i> o JSP en tiempo de ejecución. |
| <code><jsp:forward></code> | Transfiere el control a otra vista, en nuestra aplicación a otra <i>jsp</i> . |
| <code><jsp:param></code> | Permite para pasar parámetros. |

Tabla 5: JSP - Acciones

- Expresiones de lenguaje (EL): es un conjunto de abreviaturas que simplifican el acceso a las propiedades de los objetos java definidos en la página.

2.4.3 JDBC

JDBC es un API de Java que nos permite ejecutar sentencias SQL. Está formado por un conjunto de clases e interfaces programadas con en Java y está formado por dos paquetes:

El paquete `java.sql`, que contiene las clases e interfaces que permiten acceder a la funcionalidad básica de JDBC. Forma parte de la edición estándar de J2SE.

El paquete `javax.sql`, que forma parte de J2EE, en el que se incluye funcionalidad avanzada del API.

Para implementar nuestro desarrollo hemos usado el paquete de la edición estándar `java.sql`.

| Clase | Descripción |
|--------------------------|---|
| DriverManager | Proporciona el método <code>getConnection</code> con el cual introduciremos los parámetros para iniciar la conexión con la base de datos. |
| Connection | Permite establecer una sesión de conexión con la base de datos. |
| Statement | Permitirá ejecutar cualquier sentencia SQL mediante el método <code>executeQuery</code> . |
| PreparedStatement | Permite ejecutar la misma sentencia varias veces de manera más eficiente, ya que la sentencia que se ejecute quedará precompilada en el moto SQL. |
| ResultSet | Con esta clase guardaremos el conjunto de registros devuelto por la consulta ejecutada. |

Tabla 6: JDBC

2.5 MySQL

Es un sistema gestor de bases de datos relacional, multihilo y multiusuario, sus principales ventajas a grandes rasgos son: eficiencia, rendimiento, facilidad de uso y que es software libre.

Dado que en nuestra aplicación predominarán las operaciones de lectura a la base de datos con el fin de mostrar a los alumnos la información de los cursos, haremos uso de MySQL ya que nos proporciona una rápida respuesta debido al uso del motor no transaccional MyISAM.



Ilustración 6: MySQL

Por tanto, las ventajas que podemos destacar de este sistema gestor de bases de datos son las siguientes:

- Está optimizado para equipos de múltiples procesadores.
- Destaca su velocidad de respuesta en operaciones de lectura por el uso del motor no transaccional.
- Posee herramientas que ayudan a la gestión y desarrollo de la base de datos: MySQL Administrator, MySQL Query Browser y MySQL Migration Toolkit.
- Se puede utilizar como cliente-servidor o incrustado en aplicaciones.
- Buen nivel de seguridad, administración se basa en usuarios y privilegios
- Cuenta con un conjunto de tipos de datos completo.

- Se distribuyen ejecutables para numerosas plataformas diferentes: AIX, BSD, FreeBSD, HP-UX, Kurisu OS, GNU/Linux, Mac OS X, NetBSD, OpenBSD, OS/2 Warp, QNX, SGI IRIX, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64, eBD, Windows (versiones desde 95), OpenVMS20.
- La API se encuentra disponible en C, C++, Eiffel, Java, Perl, PHP, Python, Ruby y TCL.
- Soporta múltiples métodos de almacenamiento de las tablas, con prestaciones y rendimiento diferentes para poder optimizar el SGBD a cada caso concreto.
- Soporta opciones de conectividad: TCP/IP, sockets UNIX y sockets NT y ODBC.
- Al ser un software de fuente abierta, se puede descargar el código y ajustarlo a nuestras necesidades sin necesidad de obtener una licencia de pago.

2.6 Apache Tomcat

Tomcat es un contenedor de *servlets* que implementa las especificaciones de *Java Servlet* y *JavaServer Pages* (JSP).

Tomcat es un servidor web con soporte de *servlets* y JSPs. Incluye el compilador Jasper (Jasper 2 JSP Engine), que compila JSPs convirtiéndolas en *servlets*. El motor de *servlets* de Tomcat a menudo se presenta en combinación con el servidor web Apache.

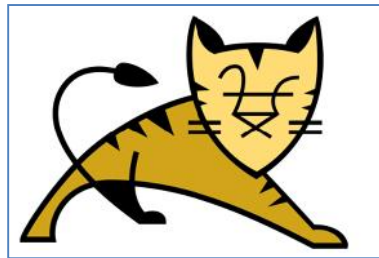


Ilustración 7: Tomcat

Hemos decidido el uso de Tomcat en nuestra aplicación web porque al ser desarrollado en Java nos permite una mejor integración con el sistema, también por su facilidad de configuración y uso y puesto que es un servidor autónomo que nos permite alto nivel de tráfico y alta disponibilidad.

Entrando en el detalle de su configuración, se basa en dos ficheros:

- El fichero de configuración principal de Tomcat (*server.xml*) que se encarga de proporcionar la configuración inicial para los componentes de Tomcat y especificar su estructura.
- El descriptor del despliegue (*web.xml*), debe estar en el directorio WEB-INF y que nos sirve para desplegar nuestra aplicación, mapeando la clase que controla los *servlet* en este *xml*.

2.7 jQuery Mobile

jQuery Mobile es una librería de Javascript optimizada para web en móviles desarrollada por el equipo del proyecto jQuery.



Ilustración 8: jQuery Mobile

Las principales características de esta librería son:

- Facilidad de uso (sintaxis) y personalización.
- Interfaz de usuario optimizada para dispositivo táctil.
- Soportada por mayor número de plataformas móviles: iOS, Android, Blackberry, WebOS, Symbian, Windows Phone, etc.
- Presentación agradable al usuario: efectos en transacción de páginas, efectos al pulsar botones, sombras y formas redondeadas.
- Compatibilidad con otros frameworks móviles.
- Mayor material de aprendizaje disponible para el desarrollador (documentación, tutoriales, foros...).

En la actualidad existen otras librerías parecidas de Javascript que ayudan al desarrollo web adaptado a móviles como:

- Sencha Touch (<https://www.sencha.com/>)
- DHTMLX Touch (<http://dhtmlx.com/touch/>)

Finalmente, por su mayor facilidad de uso y por ser una librería soportada en mayor número de plataformas nos hemos decantado por jQuery Mobile para el desarrollo de nuestra aplicación Web

Capítulo 3

Análisis del sistema

3.1 Descripción

En la etapa de análisis, se definirá con el usuario, en nuestro caso con el profesor que gestionará el contenido de la aplicación, que funcionalidades debe cubrir la aplicación web y que restricciones debe tener.

Dado que la aplicación está diseñada para dispositivos móviles cobrará importancia que el aspecto de la aplicación sea de acuerdo a las expectativas del usuario y se visualice correctamente los contenidos.

Para ello se realizará una toma de requisitos y que clasificaremos en tres tipos:

- Requisitos funcionales
- Requisitos no funcionales
- Requisitos de usabilidad

3.2 Definición de requisitos

Los requisitos obtenidos en esta fase de análisis los describiremos mediante el uso de una plantilla con los siguientes campos:

- **Identificador de requisito:** Identificador único del requisito.
- **Tipo:** Requisito funcional, no funcional o de usabilidad. Los funcionales definen lo que el sistema tiene que hacer. Los requisitos no funcionales son los que definen cómo se tienen que llevar a cabo las funciones. Y los requisitos de usabilidad que hacen referencia a la facilidad de uso de la aplicación por parte de los alumnos y profesores.
- **Origen:** quien ha solicitado que el requisito sea incluido, en nuestro caso, el profesor, alumno o propio desarrollador.
- **Necesidad:** determina la importancia que tiene dentro del sistema el cumplimiento del requisito. Su valor puede ser: esencial, opcional o deseable.
- **Prioridad:** Determina la prioridad del requisito respecto a otros. Su valor puede ser: alta, media o baja.
- **Descripción:** Definición del requisito.
- **Justificación:** Por el cual se decidió incluir este requisito.

3.3 Requisitos funcionales

Los en este apartado se describirán en detalle los requisitos funcionales de la aplicación

| Id Requisito | RF-001 | | |
|---------------|--|-----------|---------------|
| Tipo | Requisito Funcional | Origen | Desarrollador |
| Necesidad | Deseable | Prioridad | Alta |
| Descripción | <p>La página será diseñada siguiendo una plantilla de tipo 'red', en la que se definirán tres áreas básicas:</p> <ul style="list-style-type: none"> • Cabecera • Área de contenido • Pie <p>Este diseño se aplicará en todas las páginas del sitio web.</p> | | |
| Justificación | <p>El mantenimiento de la página será más fácil, puesto que todos los elementos están bien ordenados.</p> <p>Estableciendo este patrón de diseño y manteniéndolo en todo el sitio web, se facilita a los usuarios la navegación, puesto que esperarán encontrar ciertos elementos en el mismo lugar en el que estaban en páginas anteriores.</p> | | |

Tabla 7: RF-001

| Id Requisito | | RF-002 | |
|----------------------|---|------------------|----------|
| Tipo | Requisito Funcional | Origen | Profesor |
| Necesidad | Esencial | Prioridad | Alta |
| Descripción | <p>Se podrá acceder a la aplicación bajo diferentes roles, cada uno de ellos con funcionalidades distintas. Estos roles serán:</p> <ul style="list-style-type: none"> • Profesor • Alumno | | |
| Justificación | <p>Cada rol tiene funcionalidades y objetivos específicos en la aplicación. Es necesario separarlos, ofreciendo a cada rol únicamente aquellos elementos o información que necesita.</p> | | |

Tabla 8: RF-002

| Id Requisito | | RF-003 | |
|----------------------|--|------------------|----------|
| Tipo | Requisito Funcional | Origen | Profesor |
| Necesidad | Deseable | Prioridad | Media |
| Descripción | <p>La funcionalidad <i>bread crumbs</i> (“miga de pan”) en la navegación aparece en la parte superior y representa claramente la ruta de navegación recorrida hasta el momento entre los módulos, secciones y cursos</p> | | |
| Justificación | <p>Es necesario crear esta funcionalidad para que el usuario pueda saber en cualquier momento el lugar de la aplicación donde se encuentra y desde qué página ha accedido.</p> | | |

Tabla 9: RF-003

| Id Requisito | RF-004 | | |
|----------------------|--|------------------|----------|
| Tipo | Requisito Funcional | Origen | Profesor |
| Necesidad | Esencial | Prioridad | Alta |
| Descripción | Para acceder a la aplicación será necesario autenticarse. Cada usuario, ya sea alumno o profesor tendrá su nombre de usuario y contraseña para ello. | | |
| Justificación | Permitir el acceso al material didáctico a los usuarios correspondientes, ya sea alumnos para su aprendizaje o profesor para su gestión. | | |

Tabla 10: RF-004

| Id Requisito | RF-005 | | |
|----------------------|--|------------------|---------------|
| Tipo | Requisito Funcional | Origen | Desarrollador |
| Necesidad | Deseable | Prioridad | Media |
| Descripción | Las páginas deben presentar un esquema de colores y elementos de navegación similares y organizados de manera que el diseño sea consistente. | | |
| Justificación | El usuario navegará dentro de la aplicación Web por diferentes secciones y si se proporciona un diseño consistente, este sabrá que aunque ha llegado a un área nueva sigue dentro de la misma Web. | | |

Tabla 11: RF-005

| Id Requisito | RF-006 | | |
|----------------------|---|------------------|----------|
| Tipo | Requisito Funcional | Origen | Profesor |
| Necesidad | Esencial | Prioridad | Alta |
| Descripción | La aplicación permite la desconexión por parte del profesor y alumno. | | |
| Justificación | De esta forma se cerrará la sesión y se liberarán recursos. | | |

Tabla 12: RF-006

| Id Requisito | RF-007 | | |
|----------------------|---|------------------|---------------|
| Tipo | Requisito Funcional | Origen | Desarrollador |
| Necesidad | Opcional | Prioridad | Media |
| Descripción | La aplicación web permite al usuario cambiar su contraseña de acceso. | | |
| Justificación | El usuario podrá así fijar una contraseña segura y fácil de recordar. | | |

Tabla 13: RF-007

| Id Requisito | RF-008 | | |
|----------------------|---|------------------|----------|
| Tipo | Requisito Funcional | Origen | Profesor |
| Necesidad | Esencial | Prioridad | Alta |
| Descripción | El alumno tendrá acceso a los contenidos del curso de forma ordenada en módulos, secciones y cursos. | | |
| Justificación | De esta forma será más sencilla la navegación a través de contenido del curso y ayudará a su aprendizaje. | | |

Tabla 14: RF-008

| Id Requisito | RF-009 | | |
|----------------------|---|------------------|----------|
| Tipo | Requisito Funcional | Origen | Profesor |
| Necesidad | Esencial | Prioridad | Alta |
| Descripción | El profesor podrá gestionar el contenido del curso. Se permitirá añadir y activar/desactivar módulos, secciones, cursos u objetos al que tienen acceso los alumnos. | | |
| Justificación | Esta funcionalidad es necesaria para que el profesor administre el contenido del curso con agilidad a través de la aplicación. | | |

Tabla 15: RF-009

| Id Requisito | RF-010 | | |
|----------------------|--|------------------|---------------|
| Tipo | Requisito Funcional | Origen | Desarrollador |
| Necesidad | Deseable | Prioridad | Media |
| Descripción | El profesor tendrá acceso al progreso del curso por parte de sus alumnos. | | |
| Justificación | De esta forma el profesor podrá ver que partes están pendientes o completadas por cada alumno. | | |

Tabla 16: RF-010

| Id Requisito | RF-011 | | |
|----------------------|---|------------------|---------------|
| Tipo | Requisito Funcional | Origen | Desarrollador |
| Necesidad | Deseable | Prioridad | Media |
| Descripción | El alumno podrá consultar el progreso que lleva del curso. | | |
| Justificación | Así el alumno podrá tener un seguimiento de los módulos, secciones y cursos que ya haya finalizado. | | |

Tabla 17: RF-011

| Id Requisito | RF-012 | | |
|----------------------|---|------------------|---------------|
| Tipo | Requisito Funcional | Origen | Desarrollador |
| Necesidad | Esencial | Prioridad | Alta |
| Descripción | En la aplicación se mostrará al alumno los cursos completados en un color distinto. | | |
| Justificación | De esta forma el alumno verá que cursos ya ha completado, los objetos del curso estarán completados cuando el propio alumno así lo indique, ya que en la navegación por el objeto tendrá la opción de salir dejándolo pendiente o completado. | | |

Tabla 18: RF-012

| Id Requisito | RF-013 | | |
|----------------------|--|------------------|----------|
| Tipo | Requisito Funcional | Origen | Profesor |
| Necesidad | Esencial | Prioridad | Alta |
| Descripción | Los contenidos del material didáctico estarán representados en objetos <i>flash</i> (audios, teoría, ejercicios...) que se almacenarán en la carpeta de recursos del servidor web y almacenaremos la ruta en BBDD para acceder a estos objetos desde la aplicación | | |
| Justificación | De esta forma el alumno podrá acceder a la visualización de los objetos flash a través de la implementación del objeto en los <i>.jsp</i> como objeto embebido. Y el profesor podrá añadir objetos <i>flash</i> subiendo archivos <i>.swf</i> a través de una interfaz en la aplicación. | | |

Tabla 19: RF-013

3.4 Requisitos no funcionales

En este apartado se expondrán los requisitos no funcionales:

| Id Requisito | RNF-001 | | |
|----------------------|--|------------------|----------|
| Tipo | Requisito No Funcional | Origen | Profesor |
| Necesidad | Esencial | Prioridad | Alta |
| Descripción | La aplicación se ajustará a la resolución por defecto del dispositivo móvil. | | |
| Justificación | De esta forma la aplicación web se visualizará correctamente en cualquier dispositivo móvil. | | |

Tabla 20: RNF-001

| Id Requisito | RNF-002 | | |
|----------------------|---|------------------|--------|
| Tipo | Requisito No Funcional | Origen | Alumno |
| Necesidad | Esencial | Prioridad | Alta |
| Descripción | La aplicación web debe visualizarse y funcionar de forma adecuada en los navegadores web móviles (Firefox, Dolphin, Opera...) | | |
| Justificación | Así se visualizará correctamente sea cual sea el navegador que usemos en nuestro dispositivo. | | |

Tabla 21: RNF-002

| Id Requisito | RNF-003 | | |
|----------------------|--|------------------|--------|
| Tipo | Requisito No Funcional | Origen | Alumno |
| Necesidad | Deseable | Prioridad | Media |
| Descripción | Se utilizará MySQL como Gestor de Bases de Datos ya que podremos mejorar los tiempos de las consultas. | | |
| Justificación | De esta forma mejorarán los tiempos de carga de los cursos. | | |

Tabla 22: RNF-003

| Id Requisito | RNF-004 | | |
|----------------------|--|------------------|--------|
| Tipo | Requisito No Funcional | Origen | Alumno |
| Necesidad | Esencial | Prioridad | Alta |
| Descripción | La aplicación web solo será accesible desde dispositivos con conexión a internet. | | |
| Justificación | Será necesario conectar a internet para poder tener acceso al servidor donde se encuentra el material didáctico. | | |

Tabla 23: RNF-004

3.5 Requisitos de usabilidad

En este apartado describiremos los requisitos de usabilidad mediante los cuales se permite al alumno o profesor moverse de manera eficiente por cada una de las partes de la aplicación web.

| Id Requisito | | | |
|---------------|---|-----------|--------|
| RU-001 | | | |
| Tipo | Requisito de Usabilidad | Origen | Alumno |
| Necesidad | Deseable | Prioridad | Media |
| Descripción | La aplicación web permitirá acceder a los contenidos del curso en un máximo de 5 pasos (<i>clicks</i>). | | |
| Justificación | Así el alumno podrá navegar por la aplicación de forma más dinámica. | | |

Tabla 24: RU-001

| Id Requisito | | | |
|---------------|---|-----------|--------|
| RU-002 | | | |
| Tipo | Requisito de Usabilidad | Origen | Alumno |
| Necesidad | Esencial | Prioridad | Alta |
| Descripción | La aplicación debe ser robusta. | | |
| Justificación | La aplicación no debe permitir estados inconsistentes o que den lugar a fallos, así permitirá el acceso por parte del alumno o profesor en cualquier momento. | | |

Tabla 25: RU-002

| | | | |
|----------------------|--|------------------|--------|
| Id Requisito | RU-003 | | |
| Tipo | Requisito de Usabilidad | Origen | Alumno |
| Necesidad | Esencial | Prioridad | Alta |
| Descripción | Los elementos de navegación serán fácilmente identificables. Y los cursos completados se presentarán en un color distinto. | | |
| Justificación | Así el alumno puede distinguir qué elementos son navegables y además, no tendrá dudas sobre que cursos ha completado o no. | | |

Tabla 26: RU-003

Capítulo 4

Diseño e implementación

4.1 Arquitectura del sistema

La arquitectura de la aplicación se basará en el patrón Modelo – Vista – Controlador. Mediante esta separación de capas se ha facilitado el desarrollo de la aplicación web y su posterior mantenimiento.

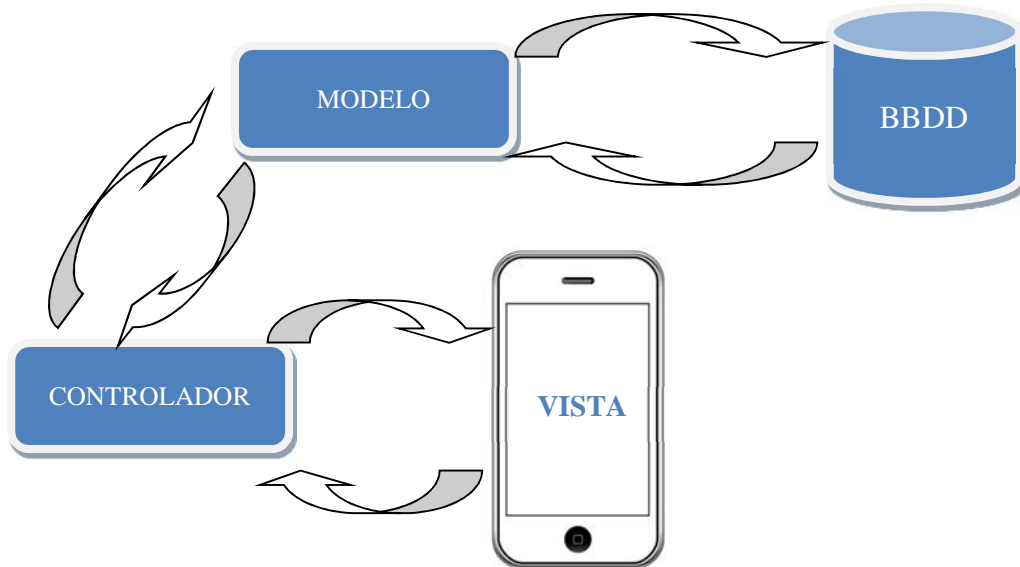


Ilustración 9: MVC

- **Modelo:** contiene los elementos de la capa de negocio, es decir representación de los datos, funcionalidad e inserción, consulta o actualización de los datos (DAOs).
- **Vista:** es la interfaz con el usuario (alumno o profesor) y representa la información que recibe a través del controlador. Se utilizarán las jsp para mostrar la información al usuario y el estilo se lo daremos utilizando la librería jQuery Mobile.
- **Controlador:** responde a eventos que realiza el usuario, es decir transforma las peticiones realizadas en la vista, en acciones que deben llevarse a cabo por el Modelo.

4.2 Base de Datos

Después de diseñar la arquitectura de la aplicación Web, es necesario diseñar el modelado de los datos para así tener la información almacenada de la forma más óptima y coherente, para ello nos hemos basado en el siguiente esquema entidad relación:

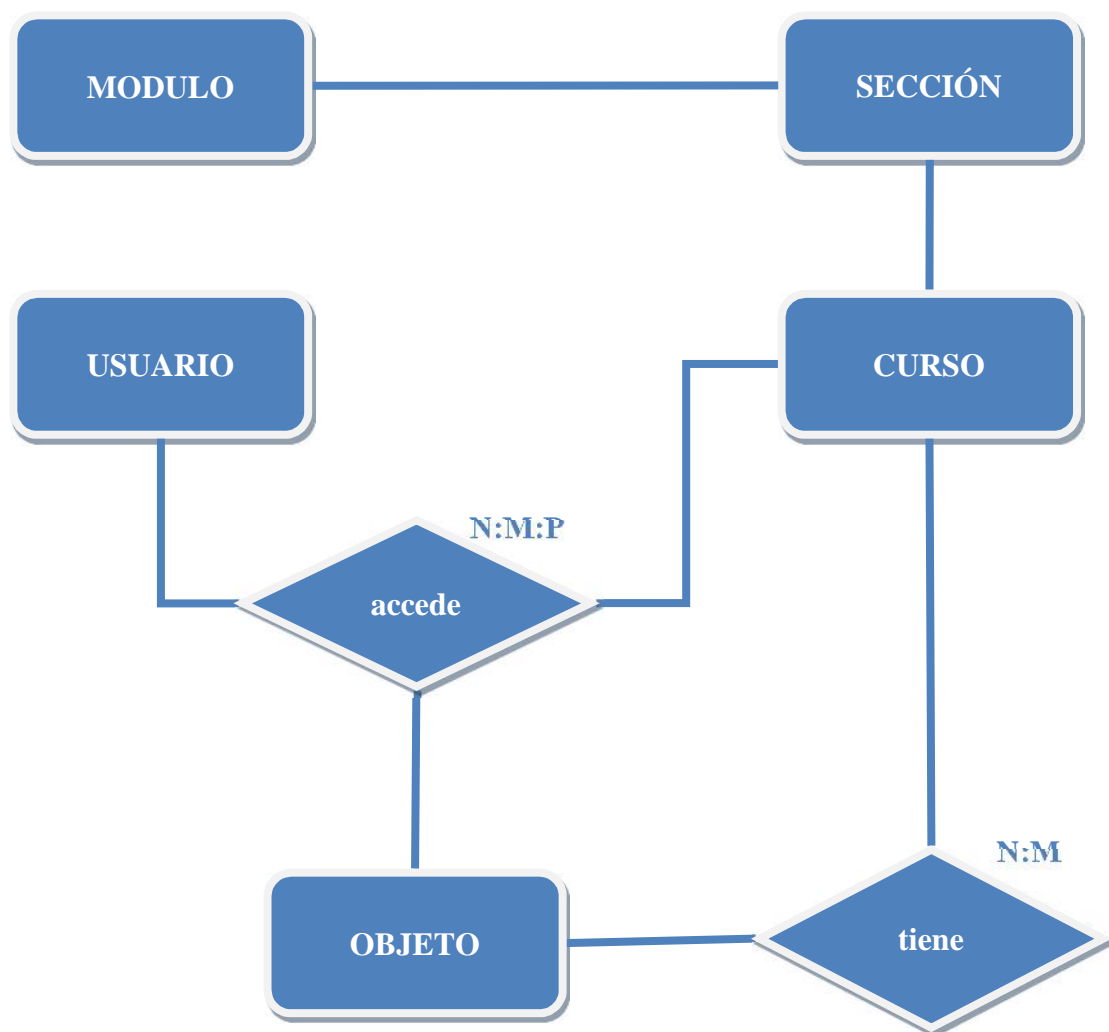


Ilustración 10: Esquema entidad relación

Partiendo del esquema entidad relación diseñaremos el modelo relacional final donde se indicarán todos los detalles (campos, claves primarias, claves ajenas...) que almacenaremos en base de datos:

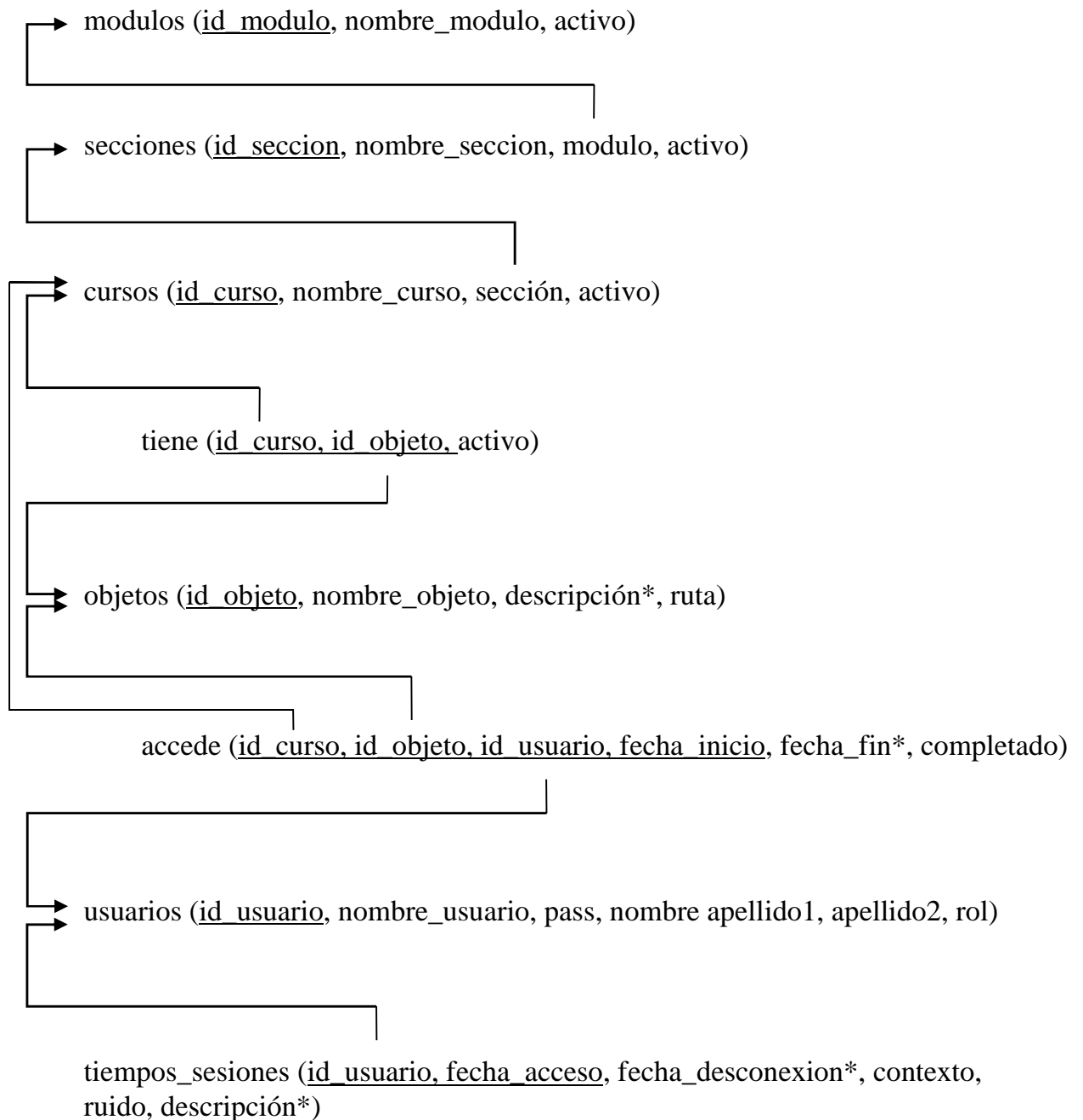


Ilustración 11: Modelo relacional

Tendremos en cuenta los siguientes supuestos semánticos:

- El campo “activo” de la tabla “modulos” puede tener los valores: Y o N
- El campo “activo” de la tabla “secciones” puede tener los valores: Y o N
- El campo “activo” de la tabla “cursos” puede tener los valores: Y o N
- El campo “activo” de la tabla “tiene” puede tener los valores: Y o N
- El campo “rol” de la tabla usuarios” puede tener los valores: Profesor o Alumno.
- El campo “completado” de la tabla “accede” puede tener los valores: Y o N
- El valor de los campos “id_modulo”, “id_seccion”, “id_curso”, “id_objeto” e “id_usuario” debe ser un valor mayor que 0.

4.3 Componentes de la aplicación

Teniendo el diseño de la base de datos y la arquitectura de la aplicación en este apartado describiremos en primer lugar cada una de las clases que hemos creado en el desarrollo:

- **Conexión BBDD:** esta clase nos permitirá la conexión con la base de datos pasando los parámetros del driver, URL, usuario y contraseña de la base de datos. Tiene las siguientes funciones:
 - obtenerInstancia
- **Usuario:** en la clase Usuario tendremos los datos del mismo y un identificador. Tiene las siguientes funciones:
 - obtenerId
 - setId
 - obtenerNombre
 - establecerNombre
 - obtenerApellido1
 - establecerApellido1
 - obtenerApellido2
 - establecerApellido2
 - obtenerUsuario
 - establecerUsuario
 - obtenerContrasenia
 - establecerContrasenia
 - obtenerActivo
 - establecerActivo
- **Modulo:** esta clase contiene la información referente a un módulo y funciones para tratar el número de objetos totales o completados que se encuentran bajo ese modulo. Tiene las siguientes funciones:
 - obtenerId
 - establecerId
 - obtenerNombre
 - establecerNombre
 - obtenerNumObjetosTotales
 - establecerNumObjetosTotales
 - obtenerNumObjetosCompletados

- establecerNumObjetosCompletados
 - obtenerActivo
 - establecerActivo
- Seccion: esta clase contiene la información referente a una sección. Tiene las siguientes funciones:
 - obtenerId
 - establecerId
 - obtenerNombre
 - establecerNombre
 - obtenerModulo
 - establecerModulo
 - obtenerActivo
 - establecerActivo
- Curso: esta clase contiene la información referente a un curso. Tiene las siguientes funciones:
 - obtenerId
 - establecerId
 - obtenerNombre
 - establecerNombre
 - obtenerSeccion
 - establecerSeccion
 - obtenerActivo
 - establecerActivo

- Objeto: esta clase contiene la información referente al objeto flash mostrado al alumno, ya sea teoría, ejercicios o audio, incluyendo la ruta donde se encuentra el objeto dentro del servidor. Tiene las siguientes funciones:
 - obtenerId
 - establecerId
 - obtenerNombre
 - establecerNombre
 - obtenerDescripcion
 - establecerDescripcion
 - obtenerRuta
 - establecerRuta
 - obtenerEstado
 - establecerEstado
 - obtenerActivo
 - establecerActivo
- Enviar datos: esta clase permitirá insertar o actualizar registros en la base de datos. Tiene las siguientes funciones:
 - enviarDatosSesion
 - actualizarDatosSesion
 - enviarDatosAccesoObjeto
 - cambiarPass
 - enviarDatosObjetoFinalizado
 - añadirModulo
 - añadirSeccion
 - añadirCurso
 - añadirObjeto
 - activarModulo
 - activarSeccion
 - activarCurso
 - activarObjeto
 - desactivarModulo
 - desactivarSeccion
 - desactivarCurso
 - desactivarObjeto

- Obtener Datos: esta clase permitirá realizar las consultas contra la base de datos para obtener las listas de cursos, sus estados, los objetos a mostrar y la totalidad de objetos pertenecientes a una sección, módulo o curso. Tiene las siguientes funciones:
 - obtenerListaUsuarios
 - obtenerListaModulos
 - obtenerListaSecciones
 - obtenerListaCursos
 - obtenerListaObjetos
 - obtenerModulo
 - obtenerSeccion
 - obtenerCurso
 - obtenerObjeto
 - objetoCompletado
 - objetoPendiente
 - contarTotalObjetos
 - contarObjetosCompletados

Para recibir las peticiones del usuario en nuestra aplicación web hemos creado una superclase que extiende la clase *HttpServlet* y que inicia los controladores web, procesa las peticiones y las dirige los generadores de las vistas (*jsp*)

La lista de controladores web que hemos creado es la siguiente:

- *Login*: este controlador valida la autenticación del usuario.
- Información de sesión: guarda la información de la sesión (lugar desde donde se conecta el usuario, nivel de ruido y descripción).
- Selección de modulo: recupera los módulos disponibles para el alumno.
- Selección de sección: recupera las secciones disponibles dado un módulo.
- Selección de curso: obtiene los cursos disponibles dado una sección.
- Selección de objeto: se obtiene el objeto seleccionado para su posterior visualización.

- Estadísticas: dado un usuario muestra el porcentaje completado de todos los módulos principales para el propio alumno o permite al profesor consultar la progresión de sus alumnos.
- Cambiar contraseña: este controlador nos llevará al formulario para que el usuario introduzca la nueva contraseña
- Comprobar contraseña: una vez el usuario rellena el formulario para cambiar la contraseña este controlador validará que el usuario ha introducido bien la contraseña actual y que la nueva contraseña introducida dos veces es correcta.
- Objeto pendiente: este controlador se encargará de registrar que el alumno ha accedido al objeto, pero lo ha dejado pendiente de finalizar.
- Objeto finalizado: registrará que el alumno ha completado el objeto.
- Añadir Módulo: mediante este controlador el profesor podrá añadir un módulo.
- Activar o desactivar módulo: se encargará de activar o desactivar el módulo para que esté o no disponible para los alumnos, se pondrá el *flag* de activo a Y o N.
- Añadir sección: mediante este controlador el profesor podrá añadir una sección indicando a que módulo pertenece.
- Activar o desactivar sección: se encargará de controlar la disponibilidad de la sección que forma parte de un módulo seleccionado, se pondrá el *flag* de activo a Y o N.
- Añadir curso: mediante este controlador el profesor podrá añadir un curso indicando a que módulo y sección pertenece.
- Activar o desactivar curso: se encargará de controlar la disponibilidad del curso que esté bajo un cierto módulo y sección seleccionado, se pondrá el *flag* de activo a Y o N, dependiendo de lo que el profesor elija.

CAPÍTULO 4: DISEÑO E IMPLEMENTACIÓN

- Añadir objeto: mediante este controlador el profesor podrá añadir un objeto indicando a que módulo, sección y curso al que pertenece y subiendo el archivo *swf* al servidor web.
- Activar o desactivar objeto: se encargará de controlar la disponibilidad del objeto para los alumnos, se pondrá el *flag* de activo a Y o N.
- *Logout*: se enviará los datos para finalizar la sesión y esta se invalidará.

Capítulo 5

Evaluación y pruebas

5.1 Entornos de pruebas

Para la validación de los componentes de la aplicación se han realizado pruebas unitarias en el propio entorno de desarrollo, es decir desde un navegador web del propio ordenador.

La aplicación web al completo se ha validado mediante pruebas de integración, donde evaluamos todos los elementos unitarios que componen la aplicación, estas pruebas se han realizado en los siguientes entornos de trabajo:

- Servidor montado sobre un ordenador *Intel Core i7* con 8GB de RAM.
- Cliente accediendo a la aplicación mediante un dispositivo móvil *Nexus 5* con *Android 5.1.1*
- Conexión a través de la misma red por wifi.
- Navegadores utilizados: *Firefox* y *Dolphin*.
- Para ello también ha sido necesaria la instalación de *Adobe Flash Player* versión 11.1 en el dispositivo móvil.

También se han realizado pruebas en otros dispositivos y navegadores con el fin de comprobar que la aplicación responde satisfactoriamente en diferentes entornos, en iOS (9.0.2) con el navegador web Safari:

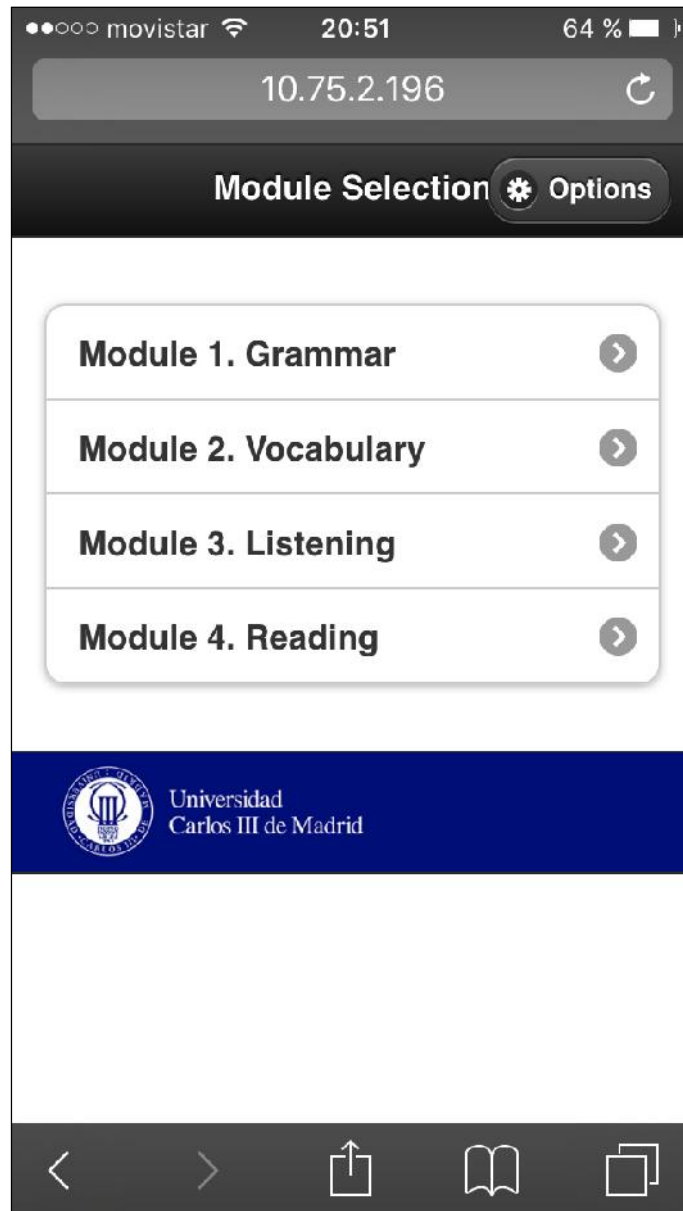


Ilustración 12: Aplicación en IOS - Safari

5.2 Definición de casos de prueba

Se han adoptado los roles de alumnos y profesores para comprobar toda la funcionalidad de la aplicación través de dispositivos móviles y se han definido los siguientes casos de prueba como necesarios para verificar la funcionalidad completa del sistema:

| | |
|---------------------------|--|
| Id Caso de prueba | CP-001 |
| Descripción | Acceder a la aplicación con un usuario del rol alumno. |
| Prerrequisitos | Exista el usuario de tipo “Alumno” insertado en la BBDD. |
| Resultado esperado | El alumno se puede identificar perfectamente y accede a la aplicación, concretamente a la pantalla de información de sesión. |
| Resultado obtenido | Correcto |

Tabla 27: CP-001

| | |
|---------------------------|--|
| Id Caso de prueba | CP-002 |
| Descripción | Acceder a la aplicación con un usuario del rol profesor. |
| Prerrequisitos | Exista el usuario de tipo “Profesor” insertado en la BBDD. |
| Resultado esperado | El profesor se puede identificar perfectamente y accede a la aplicación, concretamente a la pantalla gestión de la aplicación. |
| Resultado obtenido | Correcto |

Tabla 28: CP-002

| | |
|---------------------------|---|
| Id Caso de prueba | CP-003 |
| Descripción | El alumno puede navegar por la aplicación, por los componentes: módulos, secciones, cursos y objetos. |
| Prerrequisitos | El alumno ha accedido correctamente y existe todos los tipos de componentes activos en la BBDD. |
| Resultado esperado | La navegación es fluida y sin errores tanto de módulos hasta llegar a objetos como en sentido inverso a través de la “miga de pan”. |
| Resultado obtenido | Correcto |

Tabla 29: CP-003

| | |
|---------------------------|--|
| Id Caso de prueba | CP-004 |
| Descripción | Acceder a un objeto flash y completar el objeto accedido. |
| Prerrequisitos | Acceso del alumno y navegación hasta el objeto correcta (existiendo componentes activos). |
| Resultado esperado | El alumno puede acceder al objeto y ver el contenido del mismo, interactuar si es un ejercicio o escucharlo en caso de que sea un audio. Y en un segundo acceso al objeto ver que este está marcado como completado. |
| Resultado obtenido | Correcto |

Tabla 30: CP-004

| | |
|---------------------------|---|
| Id Caso de prueba | CP-005 |
| Descripción | Cambiar la contraseña de acceso a la aplicación. |
| Prerrequisitos | Acceso del alumno y navegación hasta el menú de opciones. |
| Resultado esperado | El alumno puede acceder al formulario donde podrá cambiar su contraseña introduciendo la antigua y la contraseña nueva dos veces. |
| Resultado obtenido | Correcto |

Tabla 31: CP-005

| | |
|---------------------------|--|
| Id Caso de prueba | CP-006 |
| Descripción | Consultar progreso del curso a medida que se completan los cursos. |
| Prerrequisitos | Acceso del alumno a la aplicación y que este complete varios cursos. |
| Resultado esperado | El alumno puede ver reflejado que el porcentaje de cursos completados va aumentando a medida que va realizando cursos. |
| Resultado obtenido | Correcto |

Tabla 32: CP-006

| | |
|---------------------------|---|
| Id Caso de prueba | CP-007 |
| Descripción | Añadir nuevo módulo en la aplicación. |
| Prerrequisitos | Conectarse a la aplicación con el rol de profesor y completar el formulario de añadir modulo. |
| Resultado esperado | Comprobar mediante el acceso de un usuario con rol de alumno que el modulo nuevo está disponible. |
| Resultado obtenido | Correcto |

Tabla 33: CP-007

| | |
|---------------------------|--|
| Id Caso de prueba | CP-008 |
| Descripción | Añadir nueva sección en la aplicación |
| Prerrequisitos | Conectarse a la aplicación con el rol de profesor y completar el formulario de añadir sección. |
| Resultado esperado | Comprobar mediante el acceso de un usuario con rol de alumno que la sección nueva está disponible. |
| Resultado obtenido | Correcto |

Tabla 34: CP-008

| | |
|---------------------------|--|
| Id Caso de prueba | CP-009 |
| Descripción | Añadir nuevo curso en la aplicación |
| Prerrequisitos | Conectarse a la aplicación con el rol de profesor y completar el formulario de añadir curso. |
| Resultado esperado | Comprobar mediante el acceso de un usuario con rol de alumno que el curso nuevo está disponible. |
| Resultado obtenido | Correcto |

Tabla 35: CP-009

| | |
|---------------------------|--|
| Id Caso de prueba | CP-010 |
| Descripción | Añadir nuevo objeto a la aplicación, subiendo el archivo .swf. |
| Prerrequisitos | Conectarse a la aplicación con el rol de profesor y completar el formulario de añadir objeto. |
| Resultado esperado | Comprobar mediante el acceso de un usuario con rol de alumno que el objeto nuevo está disponible y se puede visualizar el contenido. |
| Resultado obtenido | Correcto |

Tabla 36: CP-010

| | |
|---------------------------|--|
| Id Caso de prueba | CP-011 |
| Descripción | Activar y desactivar un módulo de la aplicación. |
| Prerrequisitos | Conectarse a la aplicación con el rol de profesor y que existan módulos tanto activos como módulos desactivados. |
| Resultado esperado | Comprobar mediante el acceso de un usuario con rol de alumno que uno de los módulos ha sido desactivado y otro activado. |
| Resultado obtenido | Correcto |

Tabla 37: CP-011

| | |
|---------------------------|--|
| Id Caso de prueba | CP-012 |
| Descripción | Activar y desactivar una sección de la aplicación. |
| Prerrequisitos | Conectarse a la aplicación con el rol de profesor y que existan secciones activas e inactivas. |
| Resultado esperado | Comprobar mediante el acceso de un usuario con rol de alumno que una de las secciones ha sido desactivada y otra activada. |
| Resultado obtenido | Correcto |

Tabla 38: CP-012

| | |
|---------------------------|---|
| Id Caso de prueba | CP-013 |
| Descripción | Activar y desactivar un curso de la aplicación. |
| Prerrequisitos | Conectarse a la aplicación con el rol de profesor y que existan cursos tanto activos como cursos desactivados. |
| Resultado esperado | Comprobar mediante el acceso de un usuario con rol de alumno que uno de los cursos ha sido desactivado y otro activado. |
| Resultado obtenido | Correcto |

Tabla 39: CP-013

| | |
|---------------------------|--|
| Id Caso de prueba | CP-014 |
| Descripción | Activar y desactivar un objeto de la aplicación. |
| Prerrequisitos | Conectarse a la aplicación con el rol de profesor y que existan objetos tanto activos como inactivos. |
| Resultado esperado | Comprobar mediante el acceso de un usuario con rol de alumno que uno de los objetos ha sido desactivado y otro activado. |
| Resultado obtenido | Correcto |

Tabla 40: CP-014

| | |
|---------------------------|---|
| Id Caso de prueba | CP-015 |
| Descripción | Ver estadísticas de un alumno seleccionado. |
| Prerrequisitos | Acceder con el usuario de tipo profesor y que exista un alumno con algún curso completado. |
| Resultado esperado | Ver a través de la consulta que el alumno ha aumentado su progreso en el curso para preparar la certificación B1. |
| Resultado obtenido | Correcto |

Tabla 41: CP-015

| | |
|---------------------------|---|
| Id Caso de prueba | CP-016 |
| Descripción | Desconectarse de la aplicación |
| Prerrequisitos | Acceder con el usuario de tipo profesor o alumno y probar a desconectarse de la aplicación. |
| Resultado esperado | El resultado visual será volver a la página de acceso. Se comprobará que la sesión se ha terminado. |
| Resultado obtenido | Correcto |

Tabla 42: CP-016

5.3 Trazabilidad de casos de prueba y resultados

En este apartado se completará una matriz de trazabilidad, en la cual se indicará la correspondencia entre los casos de pruebas definidos, y los requisitos funcionales de la especificación de requisitos.

Las filas representan cada uno de los casos de pruebas definidos, y las columnas los requisitos funcionales. Si un caso de prueba se encarga de verificar un requisito, se tendrá que señalar con una X la casilla correspondiente.

Los resultados tras la batería de pruebas y el uso en diferentes entornos han sido satisfactorios y el funcionamiento del sistema ha sido el esperado.

| | RF-001 | RF-002 | RF-003 | RF-004 | RF-005 | RF-006 | RF-007 | RF-008 | RF-009 | RF-010 | RF-011 | RF-012 | RF-013 |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| CP-001 | X | X | | X | X | | | | | | | | |
| CP-002 | X | X | | X | X | | | | | | | | |
| CP-003 | X | | X | | X | | | X | | | | | |
| CP-004 | X | | | | X | | | | | | | X | X |
| CP-005 | X | | | | X | | X | | | | | | |
| CP-006 | X | | | | X | | | | | | X | | |
| CP-007 | X | | | | X | | | | X | | | | |
| CP-008 | X | | | | X | | | | X | | | | |
| CP-009 | X | | | | X | | | | X | | | | |
| CP-010 | X | | | | X | | | | X | | | | |
| CP-011 | X | | | | X | | | | X | | | | |
| CP-012 | X | | | | X | | | | X | | | | |
| CP-013 | X | | | | X | | | | X | | | | |
| CP-014 | X | | | | X | | | | X | | | | |
| CP-015 | X | | | | X | | | | | X | | | |
| CP-016 | | | | | | X | | | | | | | |

Tabla 43: Matriz de trazabilidad

Capítulo 6

Planificación y presupuesto

6.1 Planificación

La fecha de inicio de proyecto es el 24 de noviembre de 2010 y se trabajó en el desarrollo del proyecto de forma continuada hasta la fase de implementación 1 que finalizó el 6 de junio de 2011.

La fase 2 de implementación se retomó el 17 de agosto de 2015 y el proyecto se finalizó el 8 de octubre de 2015.

En la siguiente tabla se describirán las tareas llevadas a cabo para para la realización de la aplicación web para material didáctico y sus fechas de inicio y fin.

| Actividad | Fecha inicio | Fecha fin | Duración |
|-------------------------------------|--------------|------------|----------|
| Definición del problema | 24/11/2010 | 24/11/2010 | 4h |
| Análisis | 25/11/2010 | 09/12/2010 | 56h |
| Diseño | 09/12/2010 | 12/04/2011 | 96h |
| Implementación Fase 1 | 08/02/2011 | 06/05/2011 | 118h |
| Implementación Fase 2 | 17/08/2015 | 10/09/2015 | 64h |
| Pruebas | 30/08/2015 | 10/09/2015 | 26h |
| Elaboración de documentación | 10/09/2015 | 08/10/2015 | 84h |

Tabla 44: Planificación - actividades

Una vez identificadas las actividades se procede a la creación del diagrama Gantt. En este diagrama se define el orden de ejecución de las diferentes actividades.

Fase 1:

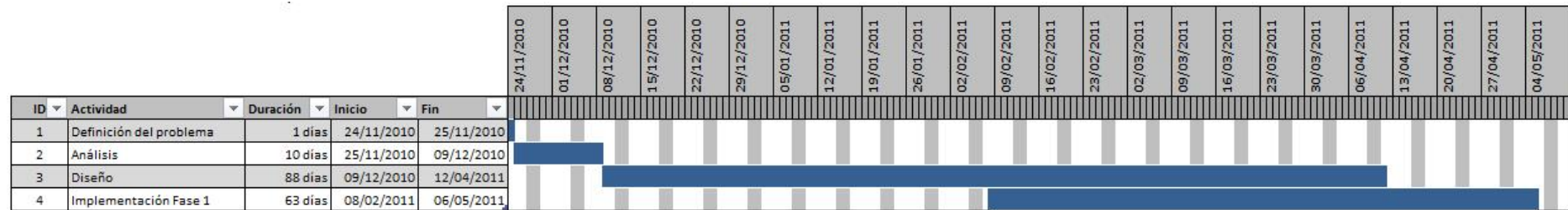


Ilustración 13: Diagrama Gantt - Fase 1

Fase 2:

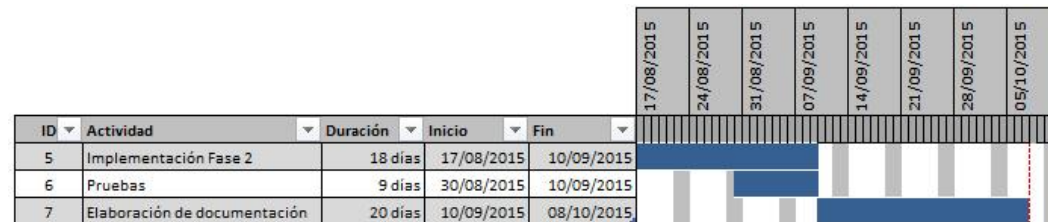


Ilustración 14: Diagrama Gantt - Fase 2

6.2 Costes del proyecto

En este apartado se describirá el coste del proyecto, para ello se tendrán en cuenta los siguientes puntos:

- Costes de personal.
- Costes de material.

6.2.1 Costes de personal

Consideraremos diferentes roles desempeñados durante las actividades identificadas en el apartado anterior para la realización de este proyecto:

| Actividad | Rol | Coste por hora | Duración | Coste |
|-------------------------------------|----------------|----------------|----------|---------|
| Definición del problema | Analista | 20 € | 4h | 80 € |
| Análisis | Analista | 20 € | 56h | 1.120 € |
| Diseño | Programador | 15 € | 96h | 1.440 € |
| Implementación Fase 1 | Programador | 15 € | 118h | 1.770 € |
| Implementación Fase 2 | Programador | 15 € | 64h | 960 € |
| Pruebas | Analista | 20 € | 26h | 520 € |
| Elaboración de documentación | Analista | 20 € | 84h | 1.680 € |
| TOTAL COSTE | 7.570 € | | | |

Tabla 45: Coste de personal

6.2.2 Costes de material

Consideraremos costes hardware y costes software:

- Los costes software no incurren ningún gasto, puesto que se ha usado programas de código abierto.
- Los costes hardware son los siguientes:

| Material | Coste |
|--------------------------------|----------------|
| PC Sobremesa Intel Core 2 Quad | 780 € |
| Portátil Lenovo Intel Core i7 | 550 € |
| Dispositivo móvil Nexus 5 | 350 € |
| Material de oficina | 25 € |
| TOTAL | 1.705 € |

Tabla 46: Coste de material

6.2.3 Coste total del proyecto

El coste total del proyecto vendrá dado de la suma de los costes de personal y de material:

| Concepto | Coste |
|--------------------|----------------|
| Costes de personal | 7.570 € |
| Costes de material | 1.705 € |
| TOTAL | 9.275 € |

Tabla 47: Coste total

Capítulo 7

Conclusiones y líneas futuras

7.1 Conclusiones

El objetivo principal de este proyecto era el desarrollo de una aplicación web en la que el alumno pudiera acceder desde cualquier lugar con su dispositivo móvil al material didáctico.

Cumplido este objetivo principal, también se ha desarrollado una interfaz para mantener actualizados los contenidos del curso por parte del profesor, pudiendo añadir, activar o desactivar contenidos dependiendo de en qué fase del curso académico se encuentre.

El acceso a la aplicación web para todos los usuarios (profesores y alumnos) se ha completado también cumpliendo los siguientes aspectos:

- Navegación ligera: las páginas no contienen excesivo contenido y cargan con rapidez.
- Correcta visibilidad: todas las paginas se adaptan a la pantalla del dispositivo móvil correctamente independientemente del sistema operativo (Android o IOS) o navegador web utilizado.
- Facilidad de uso: todas las opciones disponibles para el usuario ya sean botones o listas describen adecuadamente la acción a realizar sin lugar a confusión.

Cabe destacar que esta aplicación es reutilizable de forma sencilla para cualquier contenido didáctico que se quiera publicar con el fin de ayudar al aprendizaje de los alumnos.

Esta aplicación englobada dentro del entorno del aprendizaje nos permitirá ampliar la funcionalidad referente a este ámbito e incluso mejorar alguna de las ya implementadas, esto se procederá a describir en el siguiente punto.

Podemos concluir que el proyecto ha finalizado cumpliendo los objetivos establecidos en la fase inicial.

7.2 Líneas futuras

La aplicación ha sido diseñada de forma que permite con facilidad añadir o modificar la funcionalidad existente.

Por tanto, las posibles líneas futuras del proyecto en las que puede ser ampliado y mejorado son las siguientes:

- Añadir procesos de evaluación de contenidos para los alumnos al finalizar los módulos del curso, en forma de test que se traduciría en formularios implementados en una página *jsp* y que pudieran ser cargados por parte del profesor mediante un archivo.
- Añadir foros para cada módulo del curso, donde alumnos y profesores puedan interactuar y plasmar sus dudas y comentarios.
- Generar informe del progreso de los alumnos, actualmente está desarrollada la funcionalidad en la que el profesor puede consultar el progreso de un alumno determinado, pero por medio de este informe podrá tener una visión global del progreso de todos sus alumnos.
- Generar informes estadísticos de las sesiones para el profesor, así el profesor podrá analizar desde que lugar y con qué nivel de ruido se han realizado las sesiones y le será útil para decidir aplicar más contenido multimedia si se usa en ambientes en que el nivel de concentración es más difícil o menos.

Capítulo 8

Bibliografía

8.1 Bibliografía

1. [van Duyne et al., 2002] D. K. van Duyne, J. A. Landay, y J. I. Hong. The Design of Sites: Patterns, Principles, and Processes for Crafting a Customer-Centered Web Experience. Addison-Wesley, 2002.
2. [Javier García de Jalón · José Ignacio Rodríguez · Aitor Imaz] Servlets: Aprenda Servlets como si estuviera en Segundo. Universidad de Navarra, Abril 1999.
3. Java y J2EE: <http://java.sun.com/> (accedido diciembre de 2010, enero de 2011)
4. MySQL: <http://mysql.com/> (accedido diciembre de 2010 y septiembre de 2015)
5. jQuery Mobile. API Documentation: <http://api.jquerymobile.com/> (accedido durante la fase 2 de implementación)
6. Apache-Tomcat: <http://tomcat.apache.org/> (accedido enero y febrero de 2011)
7. Wikipedia: <http://www.wikipedia.org/> (accedido septiembre y octubre de 2015)
8. Blackboard: <http://www.blackboard.com> (accedido septiembre de 2015)
9. Moodle: <http://moodle.org> (accedido septiembre de 2015)
10. Claroline: <http://www.claroline.net> (accedido septiembre de 2015)
11. Dokeos: <http://www.dokeos.com/> (accedido septiembre de 2015)
12. Ilias: <http://www.ilias.de/docu/> (accedido septiembre de 2015)
13. Eledge Open Learning Management System: <http://eledge.sourceforge.net> (accedido septiembre de 2015)

Capítulo 9

Anexo I: Manual de usuario

9.1 Introducción

Este manual tiene por objetivo explicar el funcionamiento de la aplicación “B1 Certificate Trainer”, tanto a nivel de alumno como de profesor, ya que cada rol tiene una funcionalidad distinta dentro de la aplicación.

El manual cubrirá los requisitos necesarios para poder acceder a la aplicación, identificación de usuario, funcionalidades específicas dependiendo del rol y desconexión de la aplicación.

9.2 Requisitos necesarios

Los requisitos necesarios para poder acceder a la aplicación son los siguientes:

- Dispositivo móvil: *smartphone*, *tablet* u otro.
- Conexión a internet.
- Navegador que soporte contenido *flash*:
 - Android:
 - Firefox con Adobe Flash Player
 - Puffin Web Browser
 - Dolphin Browser
 - FlashFox - Flash Browser
 - iOS:
 - Safari
 - Puffin Web Browser
 - Skyfire Web Browser

9.3 Acceso aplicación

En este apartado describiremos como acceder e identificarnos en la aplicación.

Debemos acceder al servidor web con un navegador de internet móvil de los mencionados en el apartado anterior e ir a la siguiente dirección:

<http://localhost:8080/B1training/>

Se nos presentará una página donde podremos introducir nuestro nombre de usuario y contraseña que será facilitado previamente para acceder a la aplicación web:

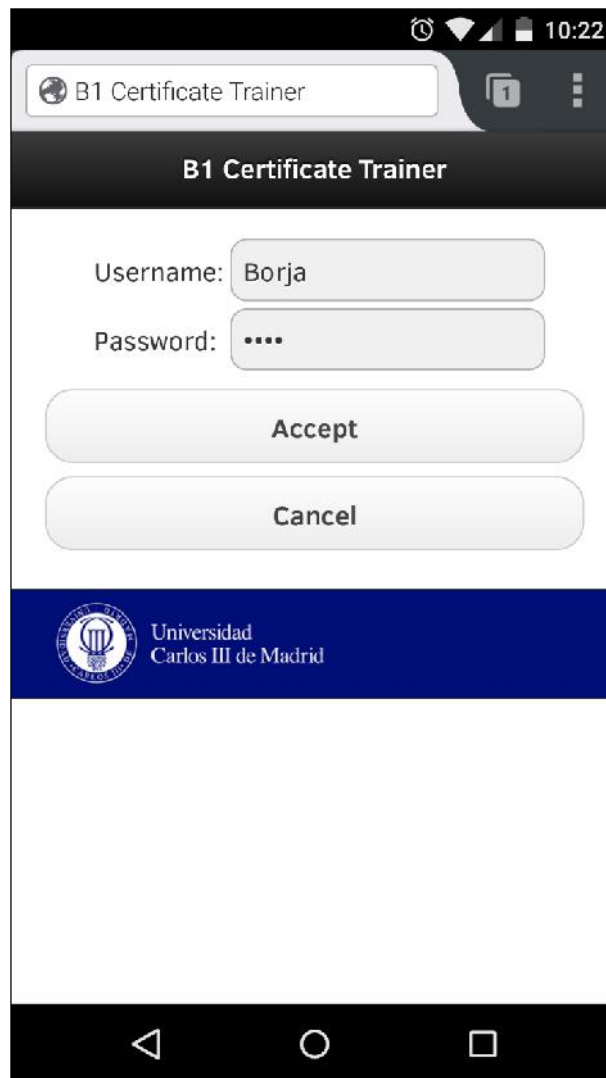


Ilustración 15: Acceso aplicación

En caso de introducir mal la contraseña, la aplicación nos permitirá introducir de nuevo los datos de acceso.

Después de acceder con el usuario y contraseña a la aplicación según el tipo de rol que desempeñemos, alumno o profesor se nos cargará una página u otra, que se describirán en los siguientes puntos.

9.4 Funcionalidades: Alumno

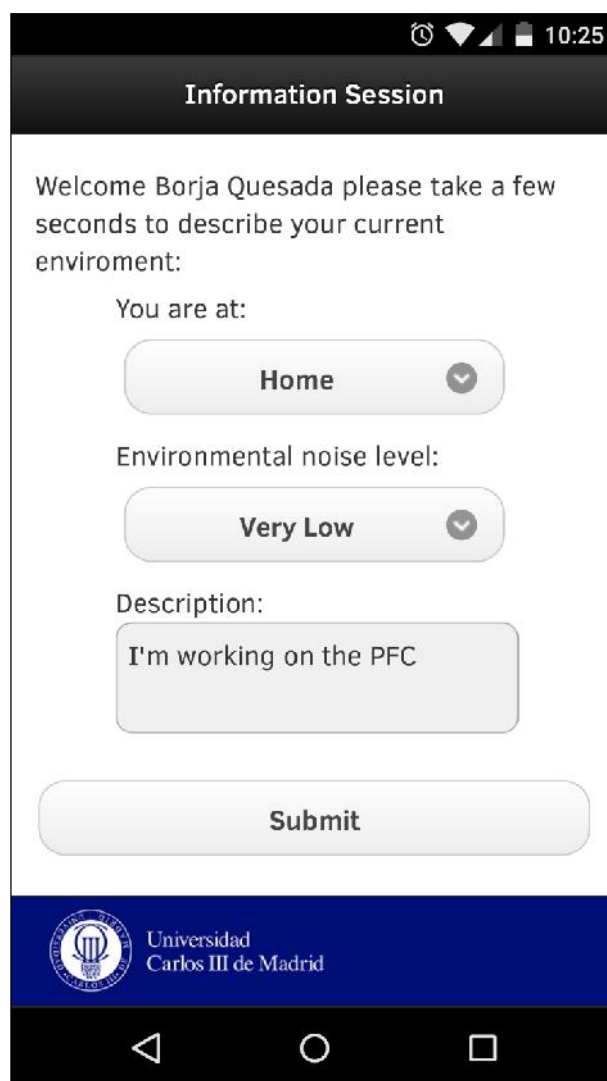
El alumno mediante la aplicación “B1 Certificate Trainer” tendrá la posibilidad de realizar el curso, en este caso particular le servirá de ayuda para poder superar la certificación nivel B1 europeo de inglés, aunque la aplicación podría mostrarse cualquier contenido didáctico.

Podrá acceder a diferentes recursos estructurado en módulos, secciones y cursos. También dispondrá de opciones para visualizar el progreso y cambiar la contraseña de acceso a la aplicación.

9.4.1 Datos de inicio de sesión

En esta página se le pedirá al alumno que indique los datos del entorno donde se encuentra realizando el curso, se le solicitará:

- Lugar: casa, trabajo, transporte, otros.
- Nivel de ruido: muy bajo, bajo, medio, alto, muy alto.
- Descripción.



The screenshot displays a mobile application interface titled "Information Session". At the top, a status bar shows the time as 10:25. The main content area begins with a welcome message: "Welcome Borja Quesada please take a few seconds to describe your current enviroment:". Below this, there are three sections for data entry: "You are at:" with a dropdown menu showing "Home", "Environmental noise level:" with a dropdown menu showing "Very Low", and "Description:" with a text input field containing "I'm working on the PFC". A large "Submit" button is positioned below these fields. At the bottom of the screen, there is a blue footer bar containing the logo and name of the "Universidad Carlos III de Madrid". The Android navigation bar is visible at the very bottom.

Ilustración 16: Datos inicio sesión

9.4.2 Realizar curso

Cuando el alumno haya enviado los datos de acceso a la aplicación podrá acceder al material del curso, la estructura del curso que se va a seguir en la aplicación es la siguiente:

- Módulos: que a su vez tendrán secciones

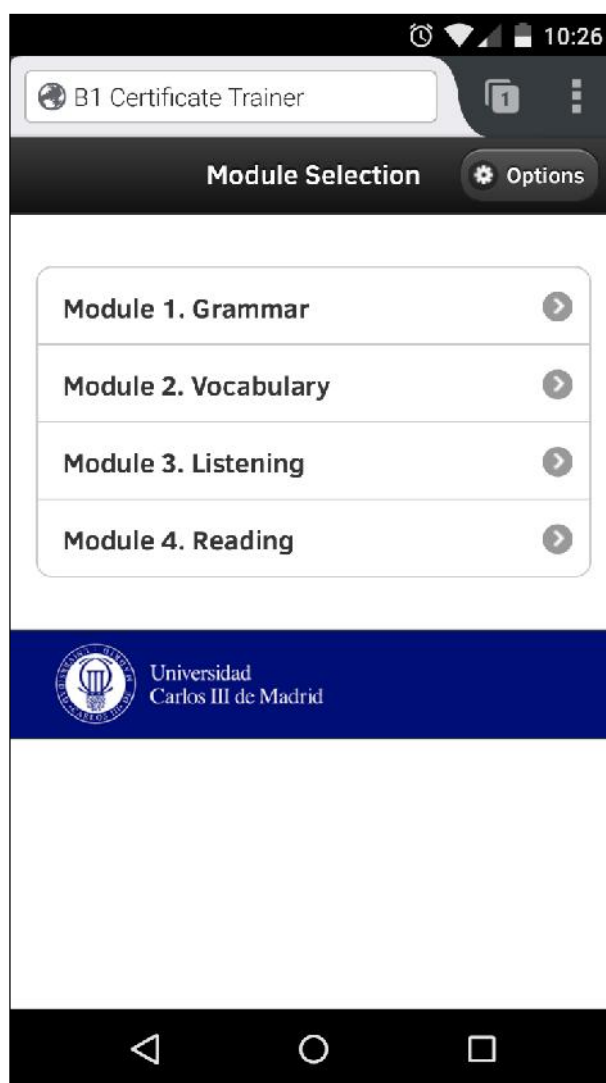


Ilustración 17: Módulos

- Secciones: que podrá ser de dos tipos:
 - Sección única donde directamente estarán los objetos flash del material didáctica (en este caso particular, material de formación para el certificado B1 de inglés)
 - Sección compuesta: que estará formada por diferentes cursos

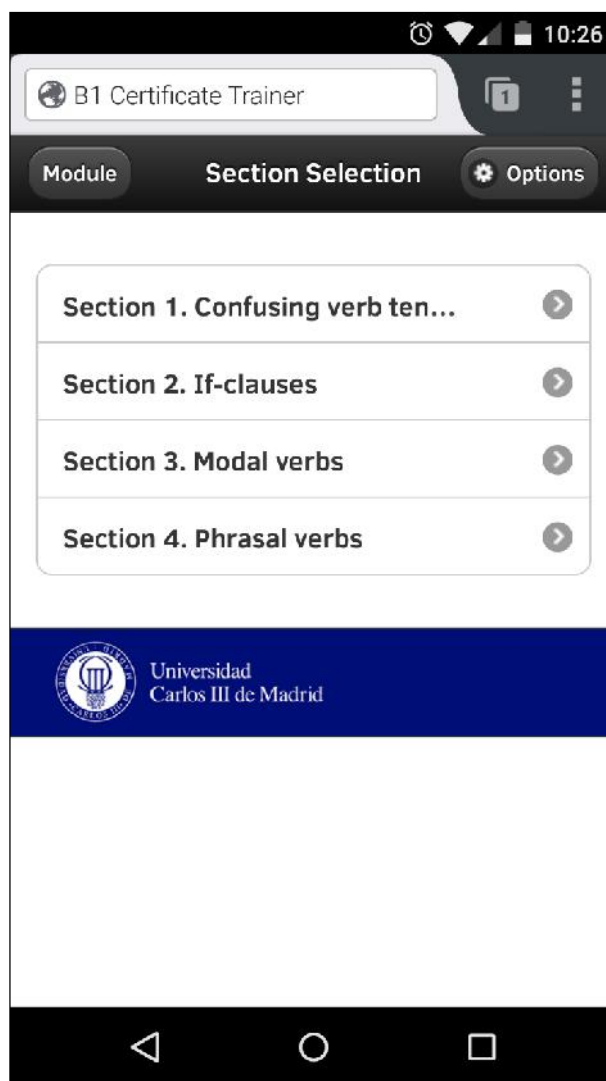


Ilustración 18: Secciones

- Cursos: que contendrán objetos.

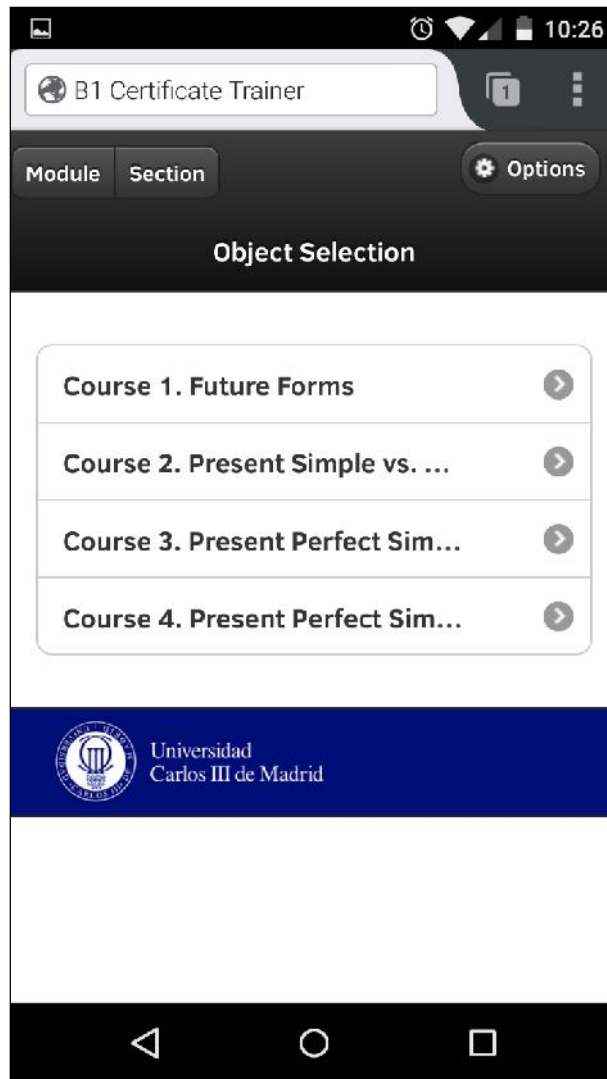


Ilustración 19: Cursos

- **Objetos:** que serán objetos flash donde se encontrará el material didáctico (información teórica, ejercicios, audios...).

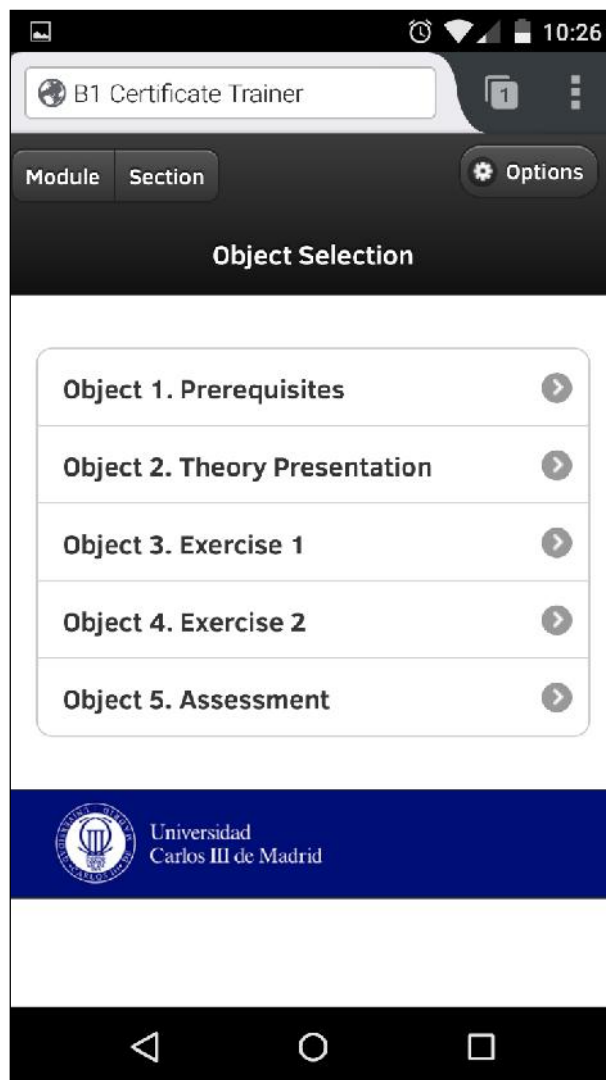


Ilustración 20: Objetos

En el momento en el que el alumno selecciona un objeto se accederá a una pantalla en la que ya se mostrará la información propia del curso embebida en un objeto flash.

Una vez accedida a esta página el alumno deberá realizar el curso o dejarlo pendiente para acabarlo en otra ocasión para ello tendrá dos botones bajo el material de estudio.



Ilustración 21: Objeto Flash

Cabe destacar que la importancia de que el alumno informe si ha acabado o no el curso, ya que si está acabado este se mostrará en un color distinto la próxima vez que acceda a la aplicación.

Podemos observar que en las pantallas de sección, curso y objeto el alumno dispone en la parte superior izquierda de un acceso a la pantalla de módulos o secciones de la que había seleccionado con anterioridad.

También dispone en la parte superior derecha de un botón para poder acceder a las opciones, las cuales serán:

- Ver progreso del curso.
- Cambiar contraseña
- Desconectarse de la aplicación

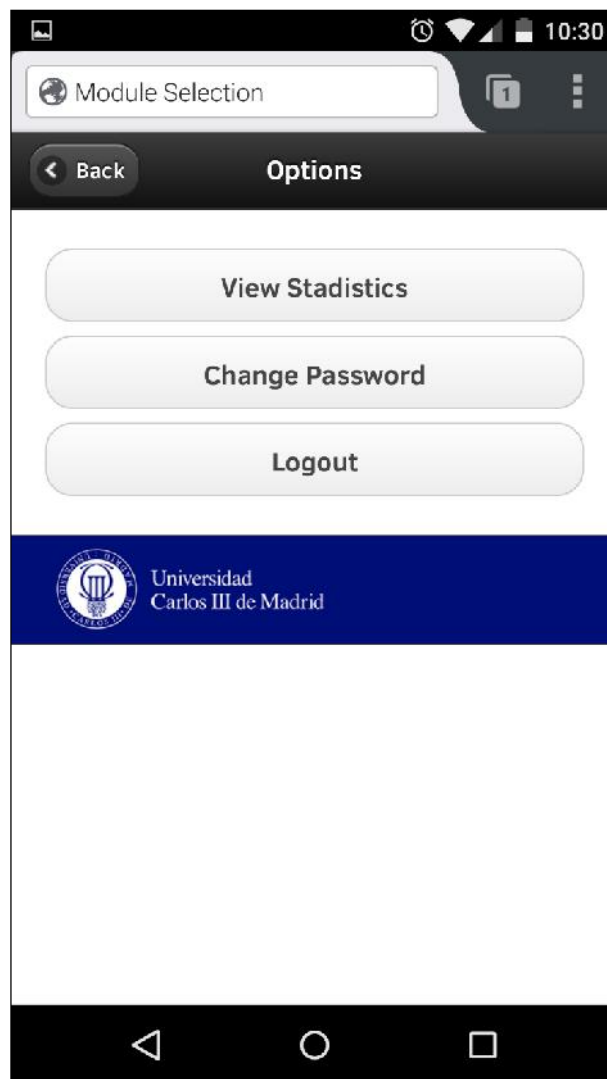


Ilustración 22: Opciones

9.4.3 Consultar progreso del curso

En esta página el alumno tendrá a su disposición un resumen estadístico del progreso del curso según el porcentaje completado de objetos, este resumen estará desagregado en cada uno de los módulos:

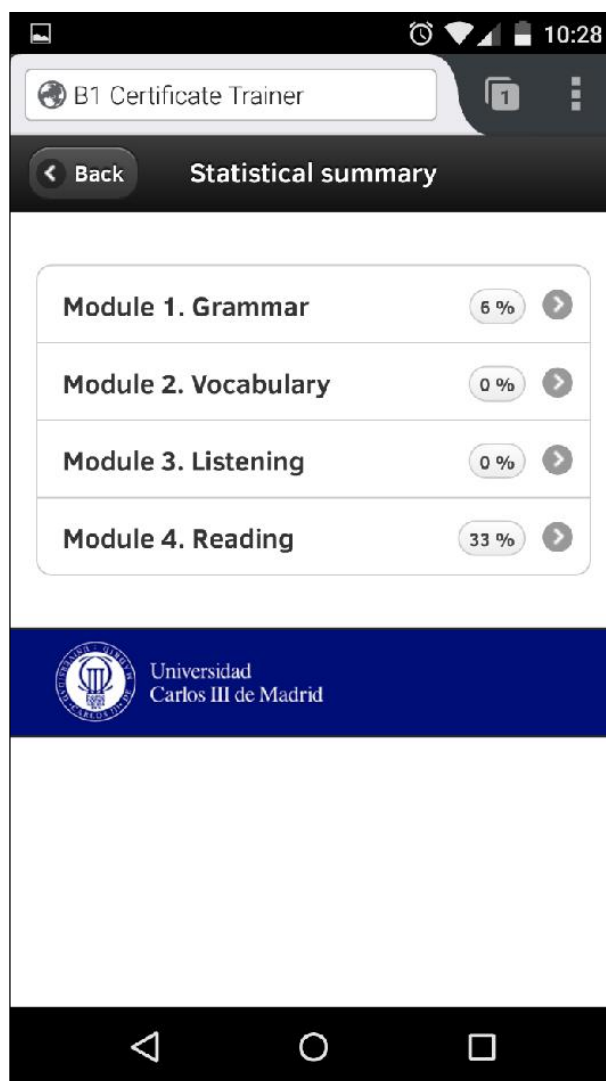
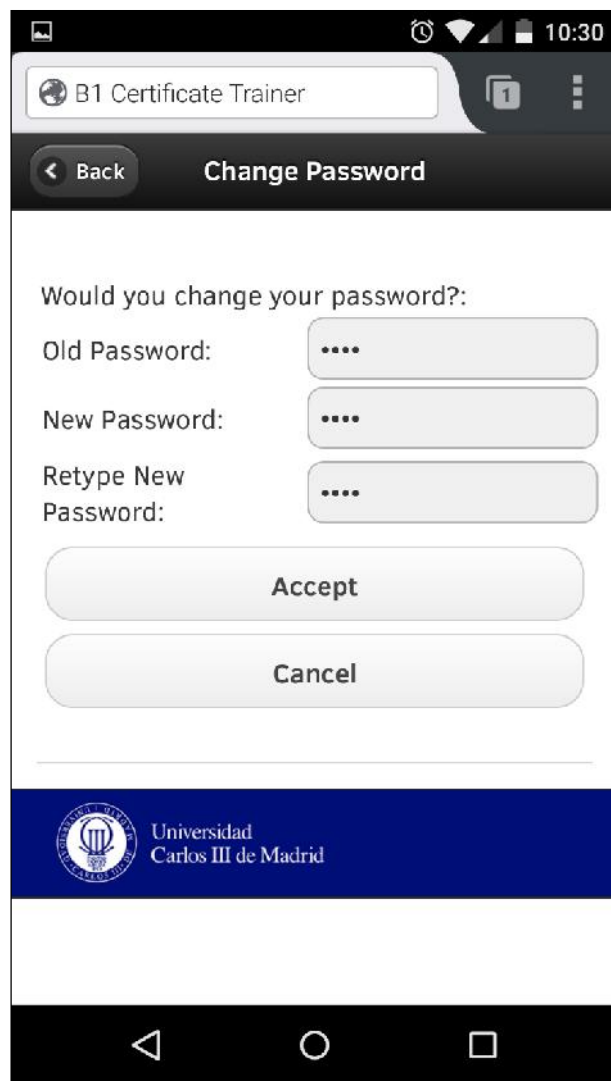


Ilustración 23: Progreso del curso

9.4.4 Cambiar contraseña

Otra de las funcionalidades disponibles al ir a las opciones que tendrá el alumno, será la de cambiar la contraseña, donde deberá informar su antigua contraseña y la nueva dos veces.



The screenshot shows a mobile application interface for 'B1 Certificate Trainer'. At the top, there's a status bar with the time 10:30 and various icons. Below it, a header bar contains a 'Back' button and the title 'Change Password'. The main content area asks 'Would you change your password?:' and features three input fields: 'Old Password:', 'New Password:', and 'Retype New Password:'. Each field is represented by a grey rounded rectangle with four dots indicating masked text. Below the input fields are two large, light-grey buttons labeled 'Accept' and 'Cancel'. At the bottom of the screen, there is a dark blue footer bar with the Universidad Carlos III de Madrid logo and name. The very bottom shows the standard Android navigation bar with back, home, and recent apps icons.

Ilustración 24: Cambio de contraseña

Una vez cambiada la contraseña la aplicación informará de ello al alumno y le redireccionará en 5 segundos a la pantalla de acceso a los módulos del curso.

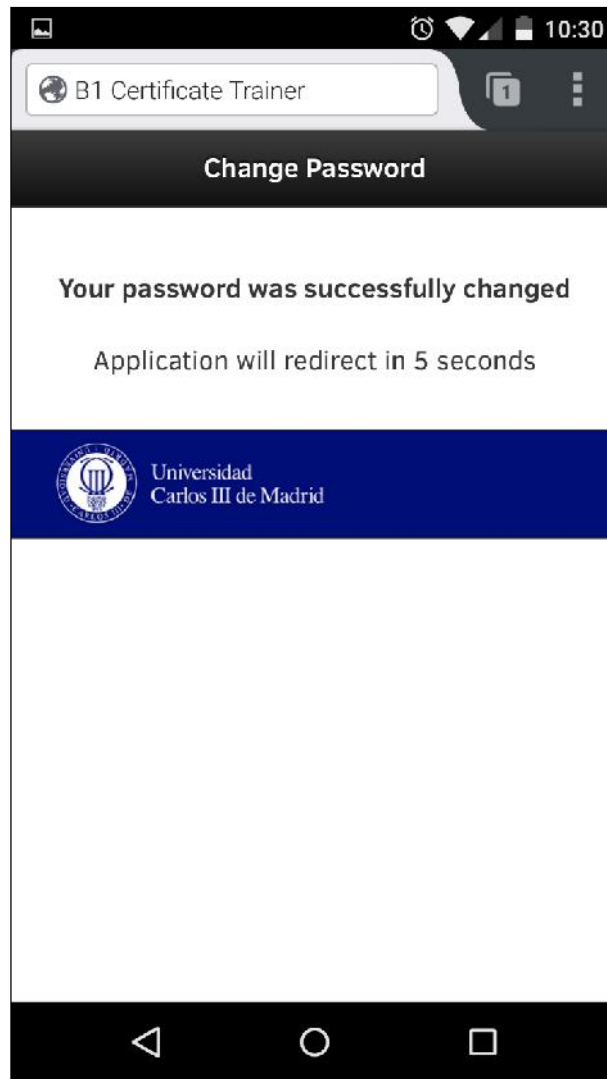


Ilustración 25: Cambio de contraseña completado

9.4.5 Desconectarse

La última opción disponible para el alumno será la de desconectarse, que cerrará correctamente la sesión del alumno y guardará la fecha y hora de su desconexión.

9.5 Funcionalidades: Profesor

El profesor se identificará de igual manera que el alumno, pero luego accederá a una pantalla de gestión del curso donde podrá gestionar y consultar datos de la aplicación.

- Gestionar módulos
- Gestionar secciones
- Gestionar cursos
- Gestionar objetos
- Consultar progreso de los alumnos

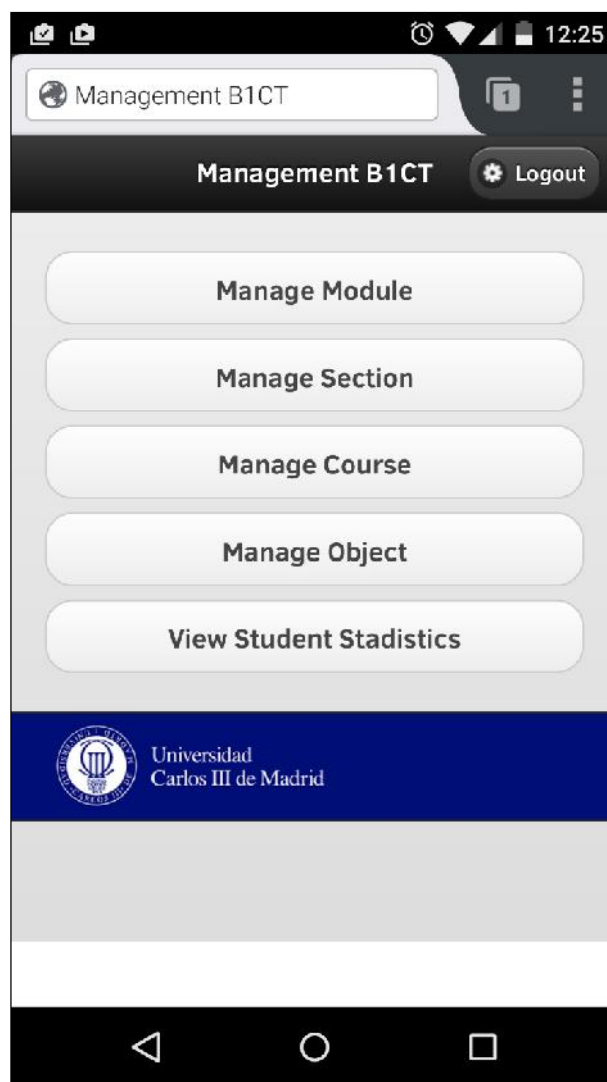


Ilustración 26: Gestión de la aplicación

En las 4 primeras opciones de gestión, el profesor tendrá la posibilidad de añadir o activar/desactivar cada uno de los componentes (modulo, sección, curso u objeto).

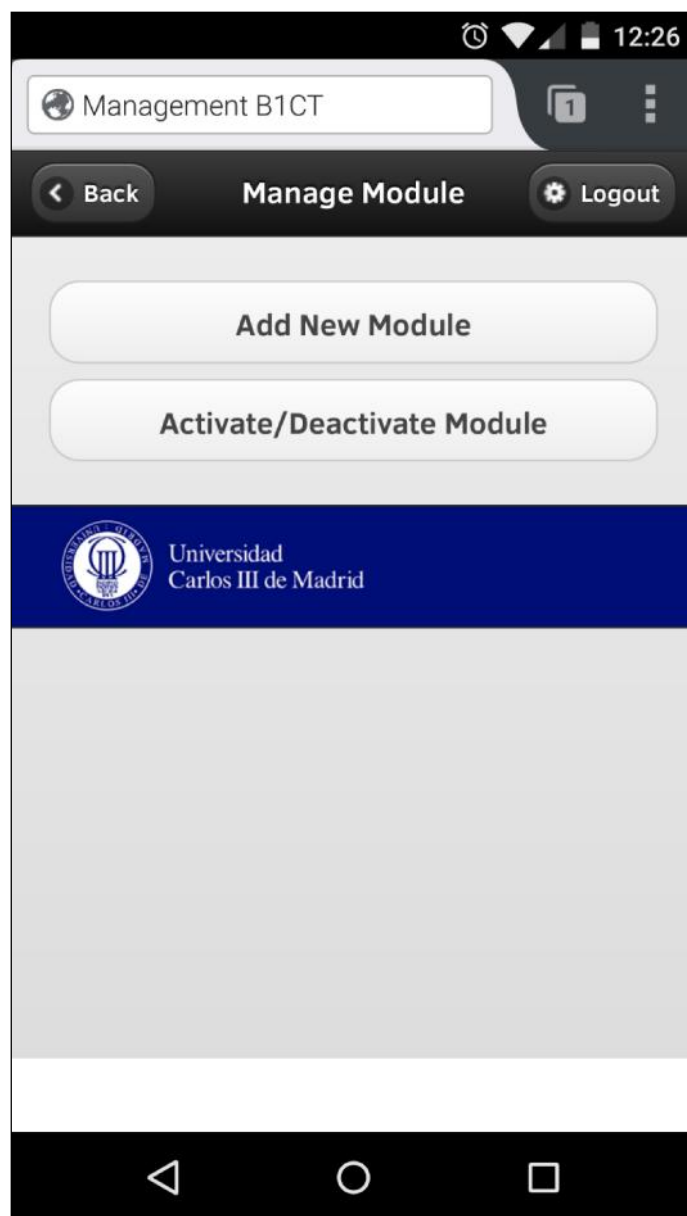


Ilustración 27: Gestión módulo

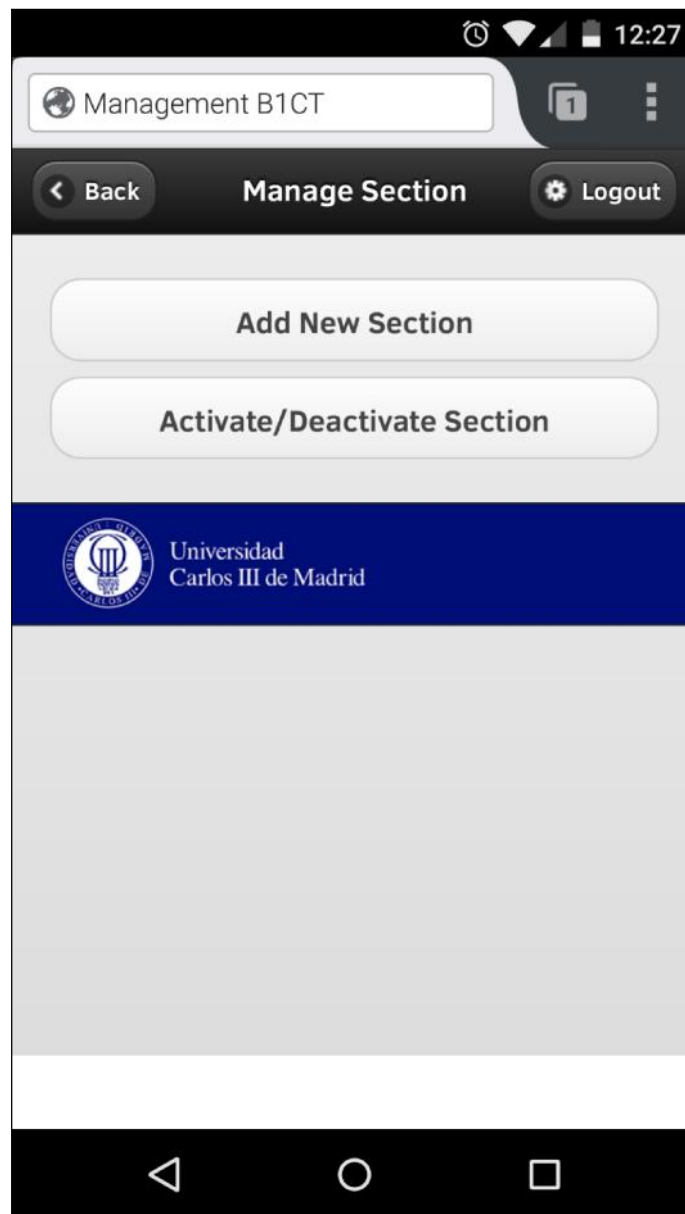


Ilustración 28: Gestión sección

Podemos destacar que desde todas estas vistas que dispone el profesor para manejar los componentes se puede volver al menú principal de gestión de la aplicación.

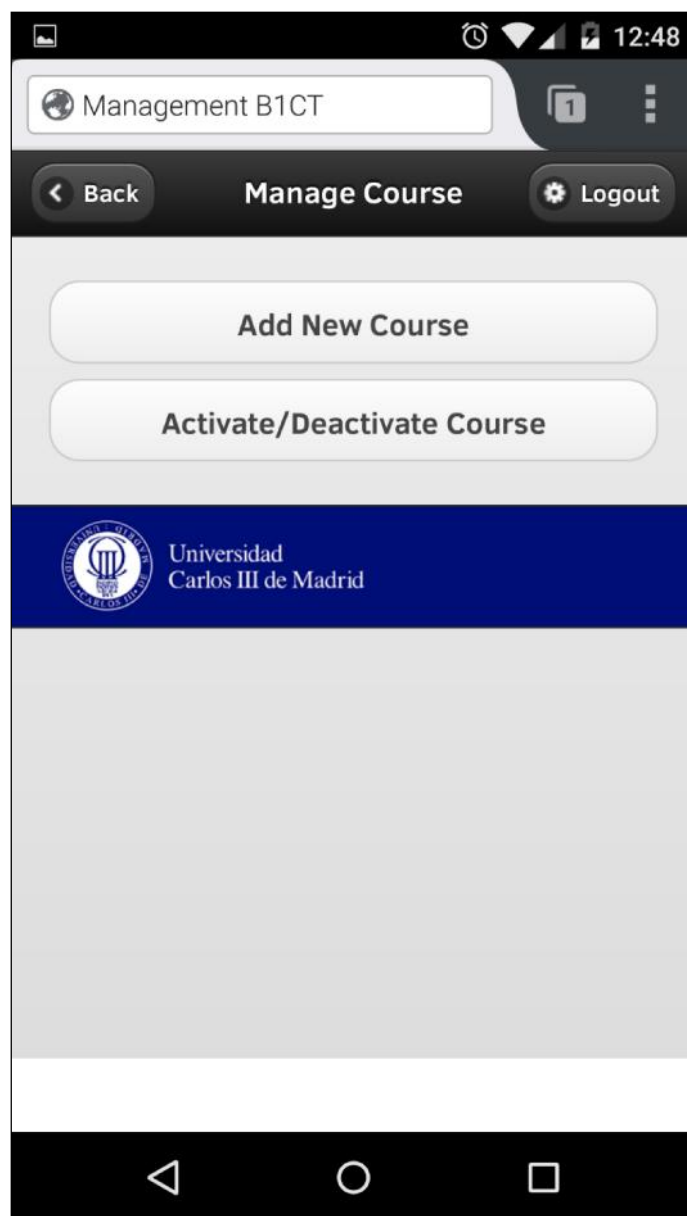


Ilustración 29: Gestión curso

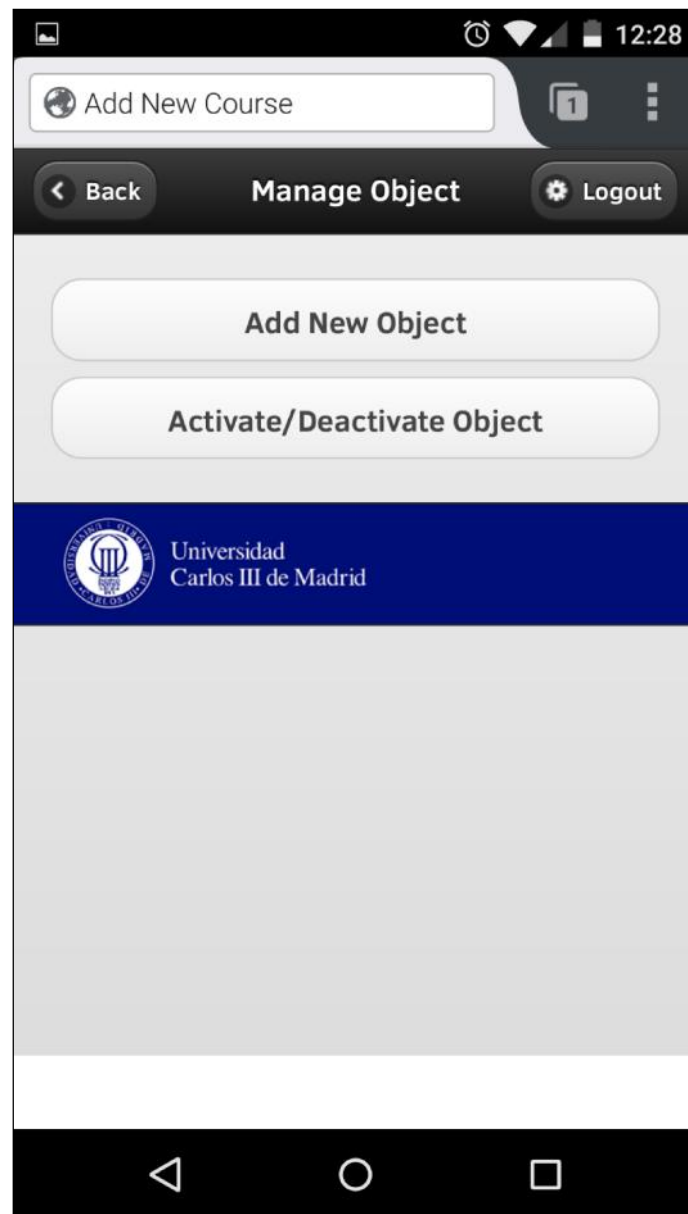


Ilustración 30: Gestión objeto

9.5.1 Añadir módulo

La aplicación dará la posibilidad al profesor de añadir un módulo informando el formulario con el nombre del módulo

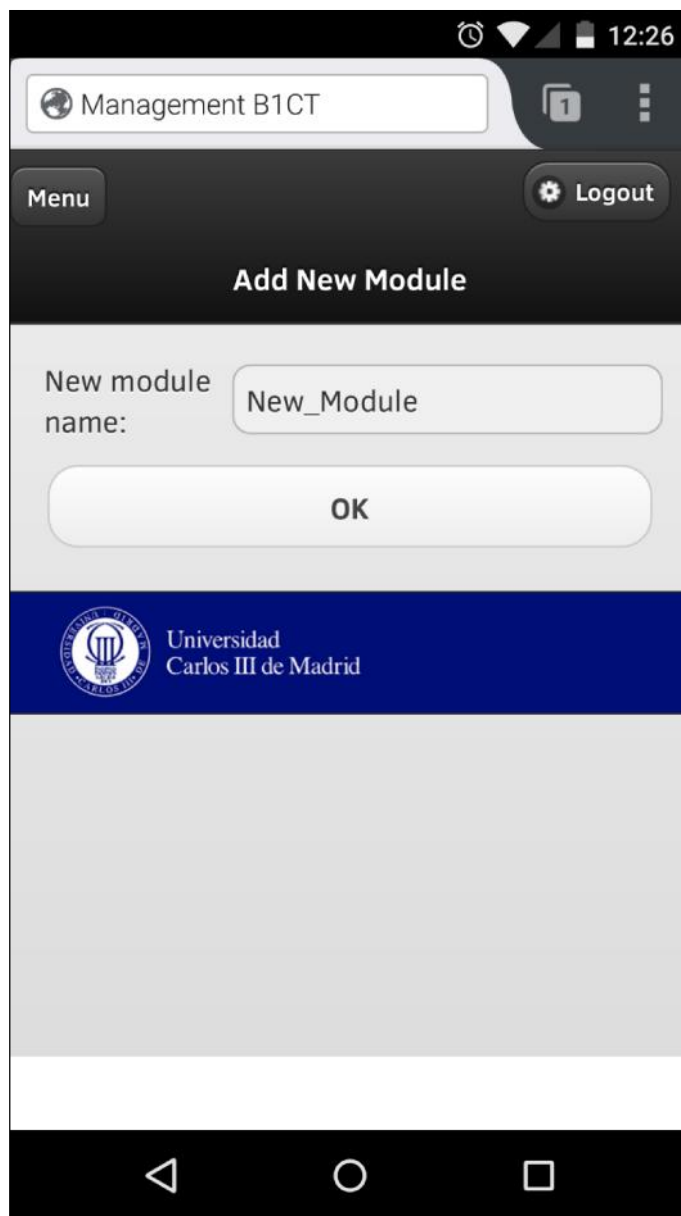
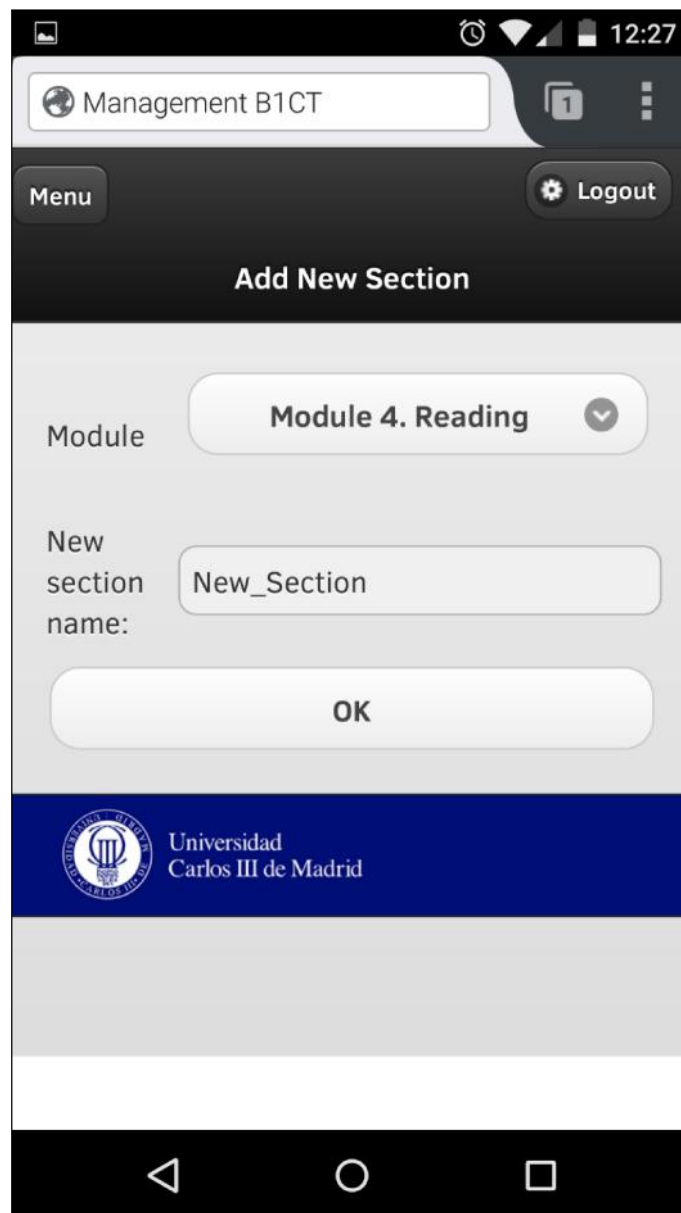


Ilustración 31: Añadir módulo

9.5.2 Añadir sección

Podrá añadir una sección, para ello deberá informar el módulo al que pertenece y el nombre de la sección.

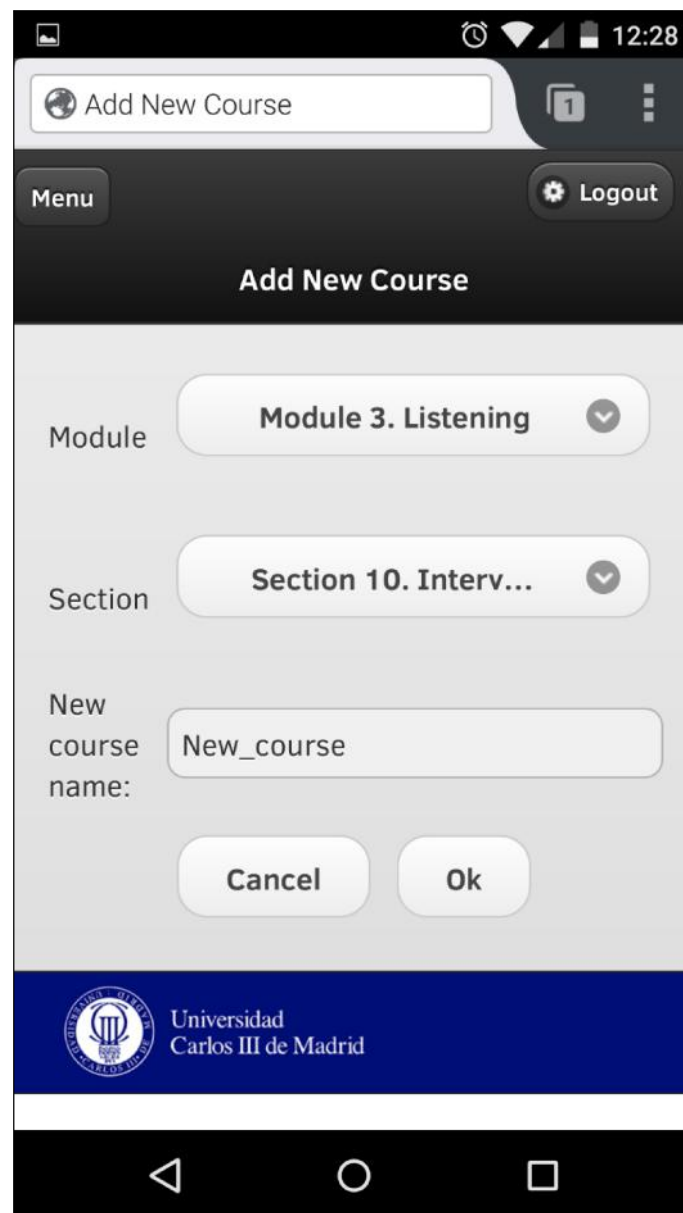


The screenshot shows a mobile application interface for 'Management B1CT'. At the top, there's a status bar with the time 12:27 and various icons. Below it, a navigation bar contains a 'Menu' button and a 'Logout' button. The main title of the screen is 'Add New Section'. The form consists of two main input fields: 'Module' with a dropdown menu currently showing 'Module 4. Reading', and 'New section name:' with a text input field containing 'New_Section'. Below these fields is a large 'OK' button. At the bottom of the screen, there is a blue banner with the logo and name of 'Universidad Carlos III de Madrid'. The very bottom shows the standard Android navigation bar with back, home, and recent apps buttons.

Ilustración 32: Añadir sección

9.5.3 Añadir curso

También le permitirá añadir un curso, de tal manera que tendrá que completar los campos del módulo, sección y nombre del curso. Estos campos se deberán completar por orden ya que cuando el profesor informe el módulo al que pertenece, se dará la opción de seleccionar una sección que forma parte de ese módulo.



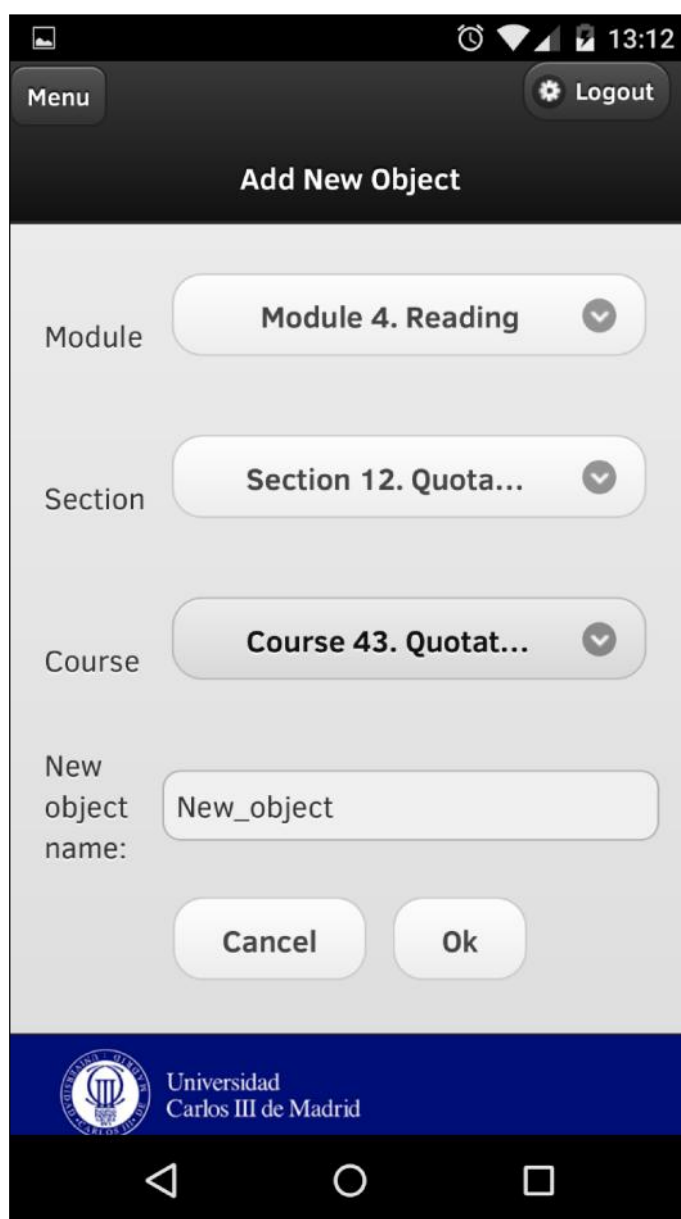
The screenshot shows a mobile application interface for adding a new course. At the top, there is a status bar with the time 12:28 and various icons. Below the status bar is a header area with a search bar containing 'Add New Course', a notification icon with the number '1', and a menu icon. The main content area is titled 'Add New Course' and contains three input fields: 'Module' with a dropdown menu showing 'Module 3. Listening', 'Section' with a dropdown menu showing 'Section 10. Interv...', and 'New course name:' with a text input field containing 'New_course'. Below these fields are two buttons: 'Cancel' and 'Ok'. At the bottom of the screen is a blue footer bar with the logo of the Universidad Carlos III de Madrid and the text 'Universidad Carlos III de Madrid'. The bottom of the screen also shows the Android navigation bar with back, home, and recent apps buttons.

Ilustración 33: Añadir curso

9.5.4 Añadir objeto

El profesor dispondrá de un formulario para poder rellenar los campos referentes a la creación de un objeto.

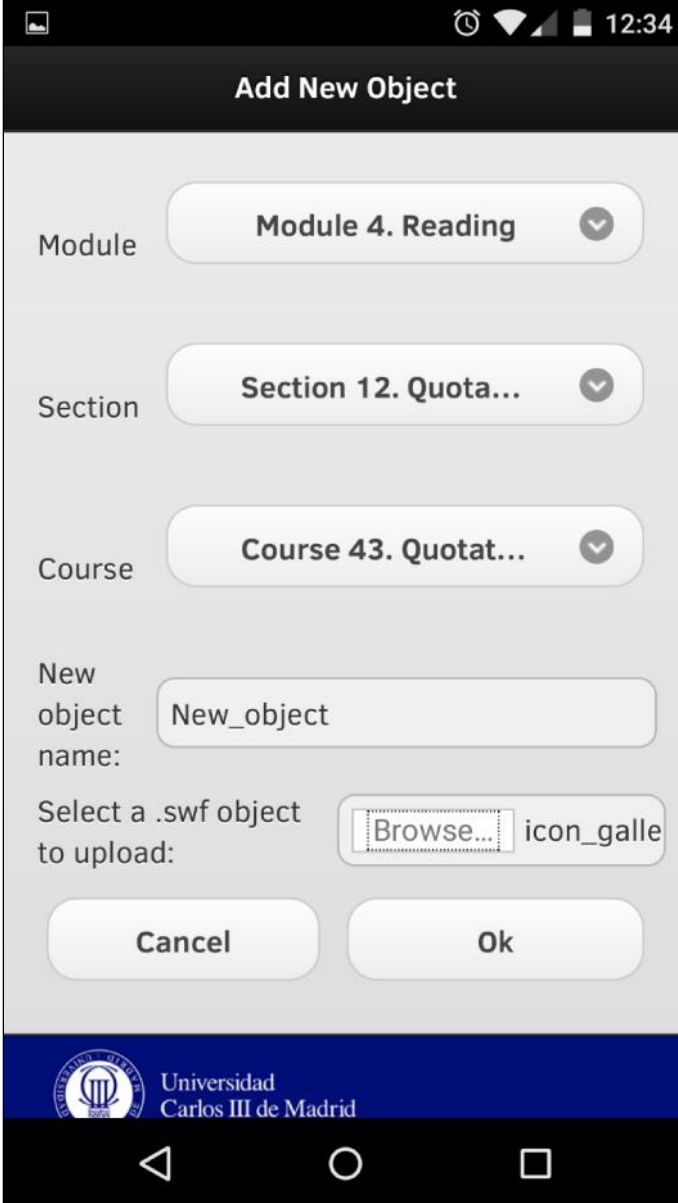
Para ello y por orden de aparición en la página deberá de informar, el modulo, la sección y el curso al que pertenece el objeto nuevo a añadir. Será necesario seguir el orden de aparición en la página ya que los campos se cargarán dinámicamente de acuerdo con la selección del campo anterior. Es decir, los cursos cargarán de acuerdo a una sección y las secciones de acuerdo al módulo seleccionado.



The screenshot shows a mobile application interface for adding a new object. The top bar is dark with a 'Menu' button on the left and a 'Logout' button on the right. The main title 'Add New Object' is centered. The form consists of three dropdown menus: 'Module' (selected: 'Module 4. Reading'), 'Section' (selected: 'Section 12. Quota...'), and 'Course' (selected: 'Course 43. Quotat...'). Below these is a text input field for 'New object name:' with the value 'New_object'. At the bottom are 'Cancel' and 'Ok' buttons. The footer shows the logo and name of 'Universidad Carlos III de Madrid'.

Ilustración 34: Añadir objeto

Al pulsar OK en el formulario anterior se habilitará el campo para poder subir el objeto flash que corresponda (con extensión *.swf*). De esta forma al pulsar OK en este segundo formulario se creará un objeto nuevo asociado a los componentes anteriormente indicados, con el nombre correspondiente y se subirá a la carpeta correspondiente del servidor web.



Add New Object

Module **Module 4. Reading** ▼

Section **Section 12. Quota...** ▼

Course **Course 43. Quotat...** ▼

New object name:

Select a .swf object to upload: icon_galle


 Universidad Carlos III de Madrid

Ilustración 35: Añadir objeto - Subir fichero

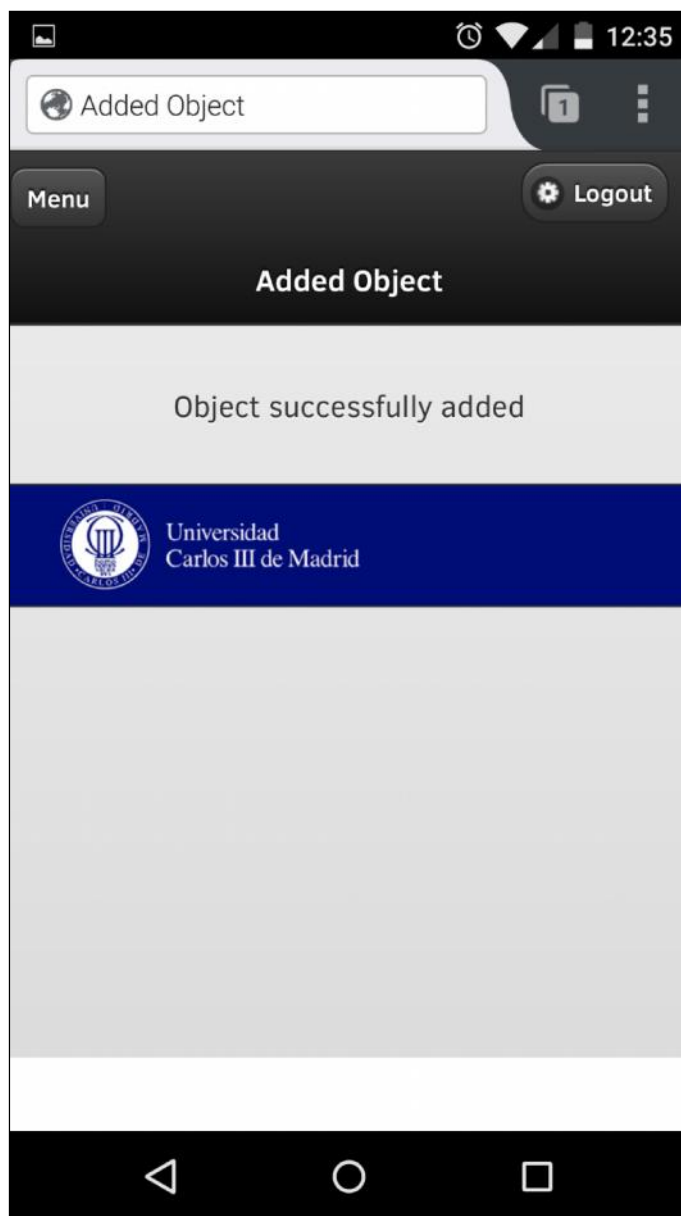


Ilustración 36: Objeto añadido

Podemos observar que en todas las páginas de añadir componentes tenemos el botón “Cancelar” el cual nos servirá para poder dejar en blanco todos los campos del formulario.

También se puede observar que existe la posibilidad de volver al menú de gestión o desconectarnos desde cualquier página de gestión de los componentes.

9.5.5 Activar/Desactivar componente

En esta sección de la aplicación podremos desactivar o activar módulos, secciones, cursos u objetos.

Se nos mostrará el listado de componentes con su estado actual y podremos activar o desactivarlos seleccionando los *check* correspondientes.

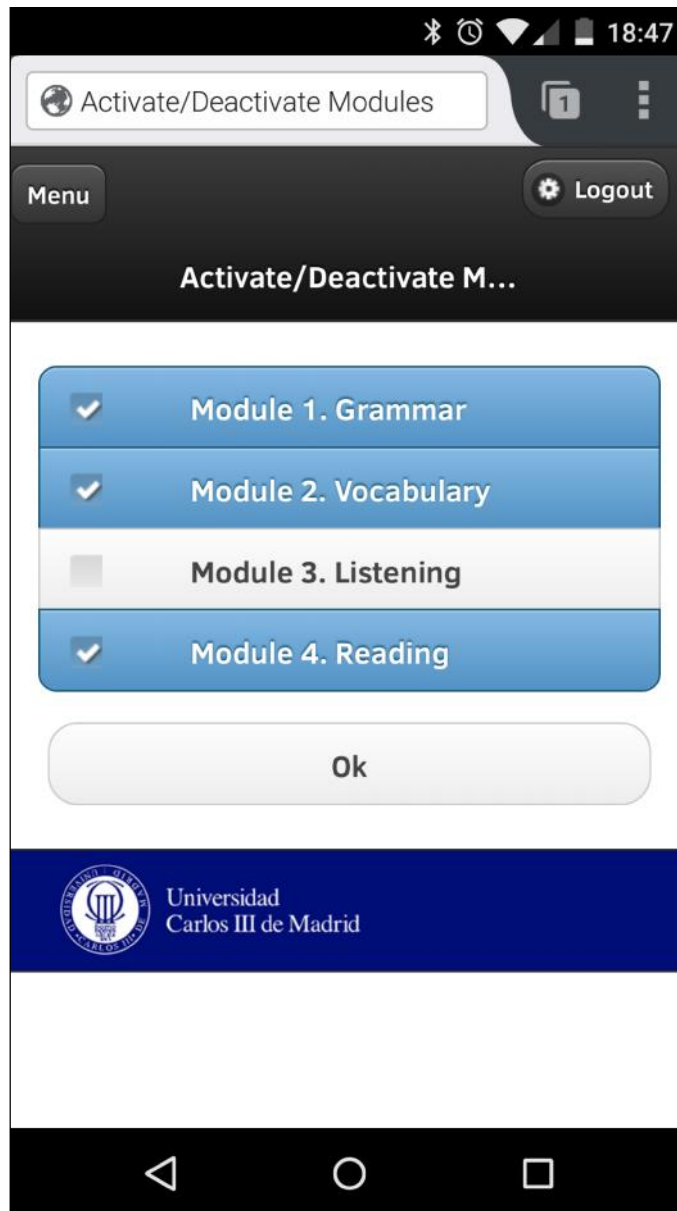


Ilustración 37: Activar/Desactivar componente

9.5.6 Consultar progreso alumno

El profesor también podrá ver el progreso de cada uno de sus alumnos, seleccionado un determinado alumno en la siguiente pantalla:

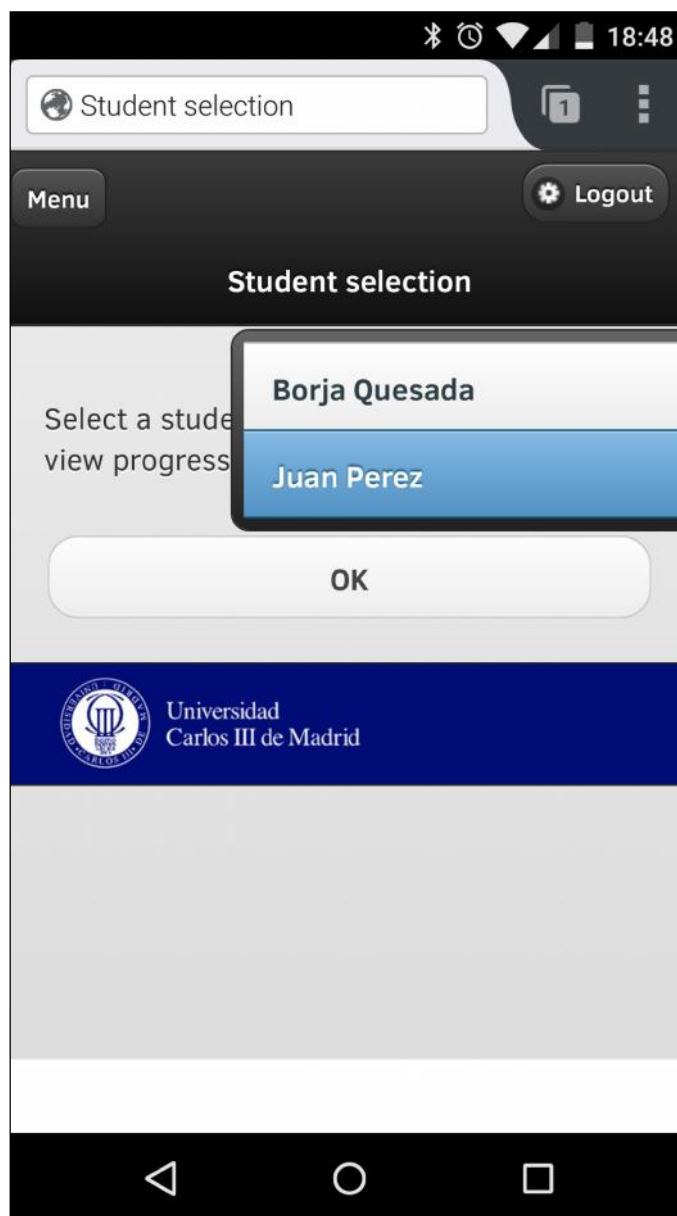


Ilustración 38: Seleccionar alumno

De esta forma la aplicación nos mostrará el progreso del alumno que se haya seleccionado:

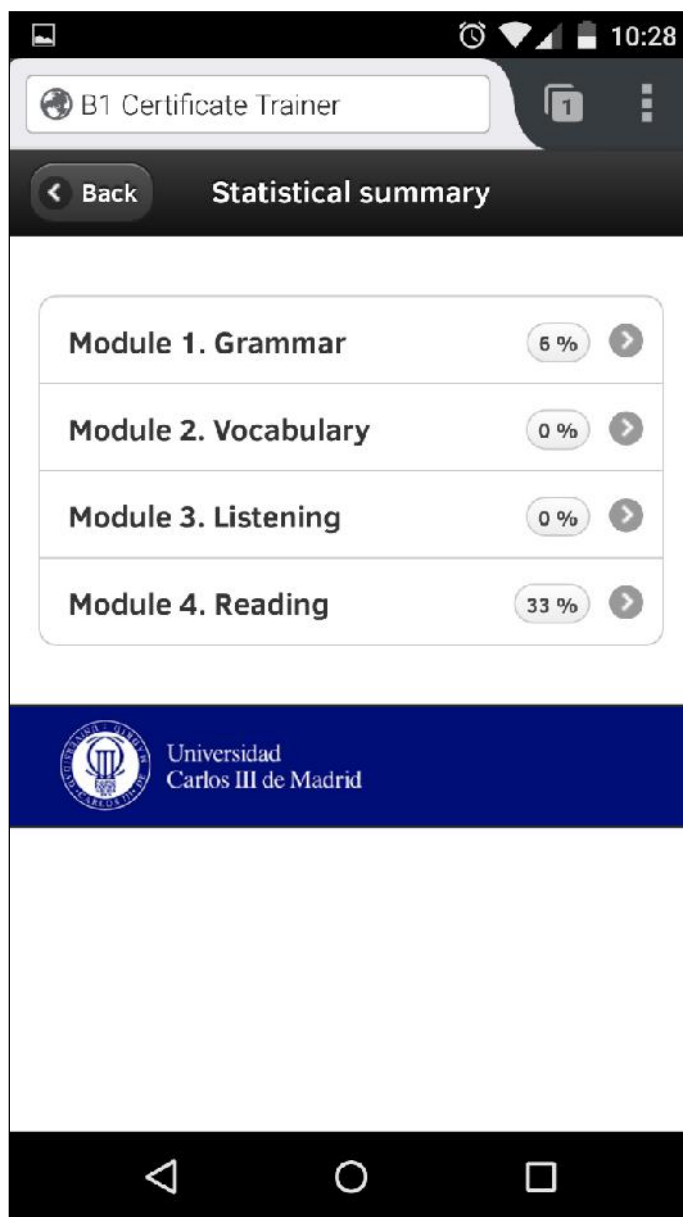


Ilustración 39: Consulta progreso alumno

9.5.7 Desconectar de la aplicación

Todas las pantallas accedidas por el rol del profesor tendrán en la parte superior derecha un botón “Logout” que permitirá desconectarse de la aplicación.